

بسم الله الرحمن الرحيم، والحمد لله رب العالمين، والصلاة والسلام على خاتم الأنبياء والمرسلين، وعلى آله وصحبه وسلم، أما بعد

السلام عليكم ورحمة الله وبركاته  
أقدم لكم شرحاً عن برنامج التشفير الجديد والذي يقوم بتشفير وفك التشفير للرسائل بين الإخوة في الجماعات الجهادية -بارك الله فيها-، وإلى الإخوة الذين يريدون الإتصال فيما بينهم ويريدون تشفير الكلام، وسوف نذكر نقاط ضعف البرنامج، وسوف نذكر إيجابياته بالتفصيل الكامل، حتى يسهل التعامل مع هذا البرنامج، وإن قدر الله ستكون هناك المزيد من البرامج، أو تطويرات في البرامج إن شاء الله رب العالمين.

أولاً البرنامج معمول بلغة الجافا مفتوحة المصدر، بإستعمال برنامج نت-بينز، ويمكنكم تنزيل البرنامج من الموقع الرسمي للبرنامج

<http://netbeans.org/>

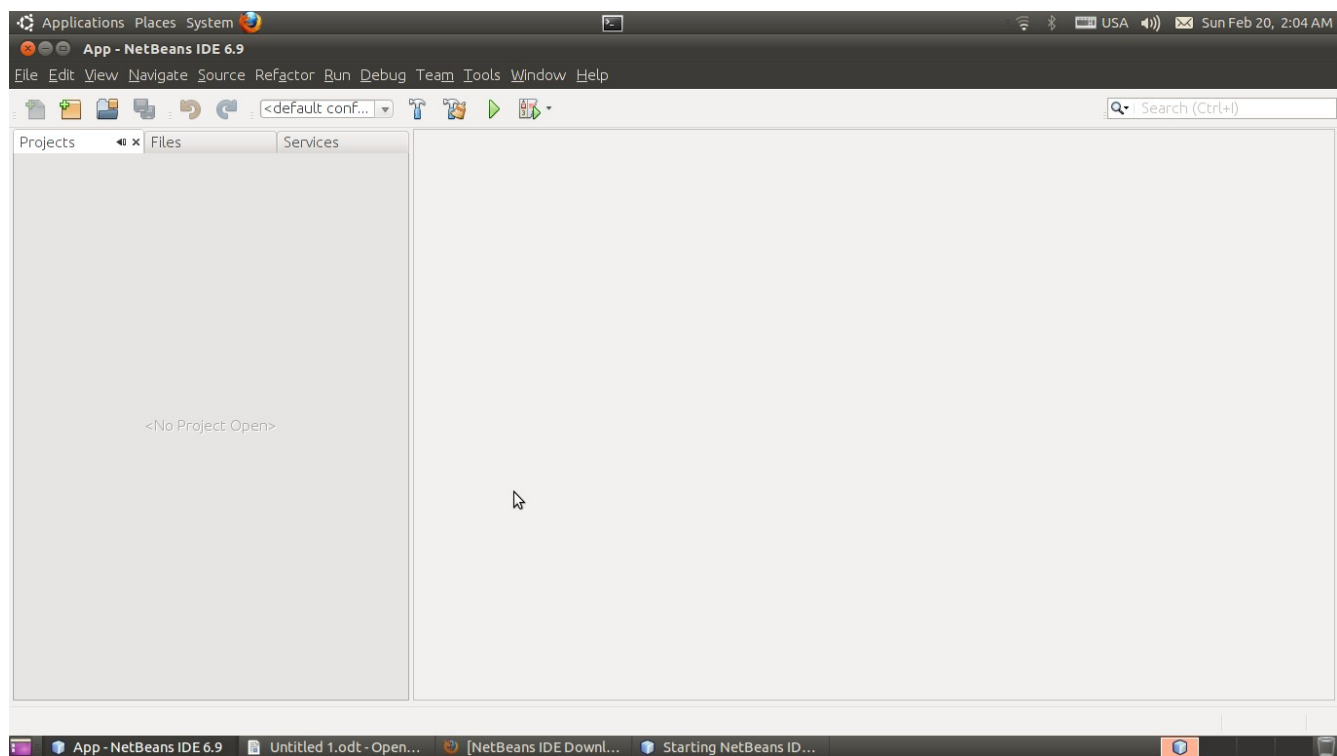
لمستخدمي توزيع أوبنتو/توزيع سيلي الإسلامية نظام لينكس: يُمكنكم تنزيلها عن طريق (مركز برمجيات أوبنتو) والبحث عن إسم البرنامج في مربع البحث  
(لمستخدمي توزيع فيدورا/توزيع أعجوبة نظام لينكس: يُمكنكم تنزيلها عن طريق (مدير الحزم add/remove Software

والبحث عن إسم البرنامج في مربع البحث  
لمستخدمي نظام ويندوز (رجاءاً غير نظامك فينظام ويندوز ليس آمناً) يُمكنكم تنزيلها بالدخول إلى الموقع والضغط على تنزيل، ومن ثم سوف تأتي لكم هذه الصورة

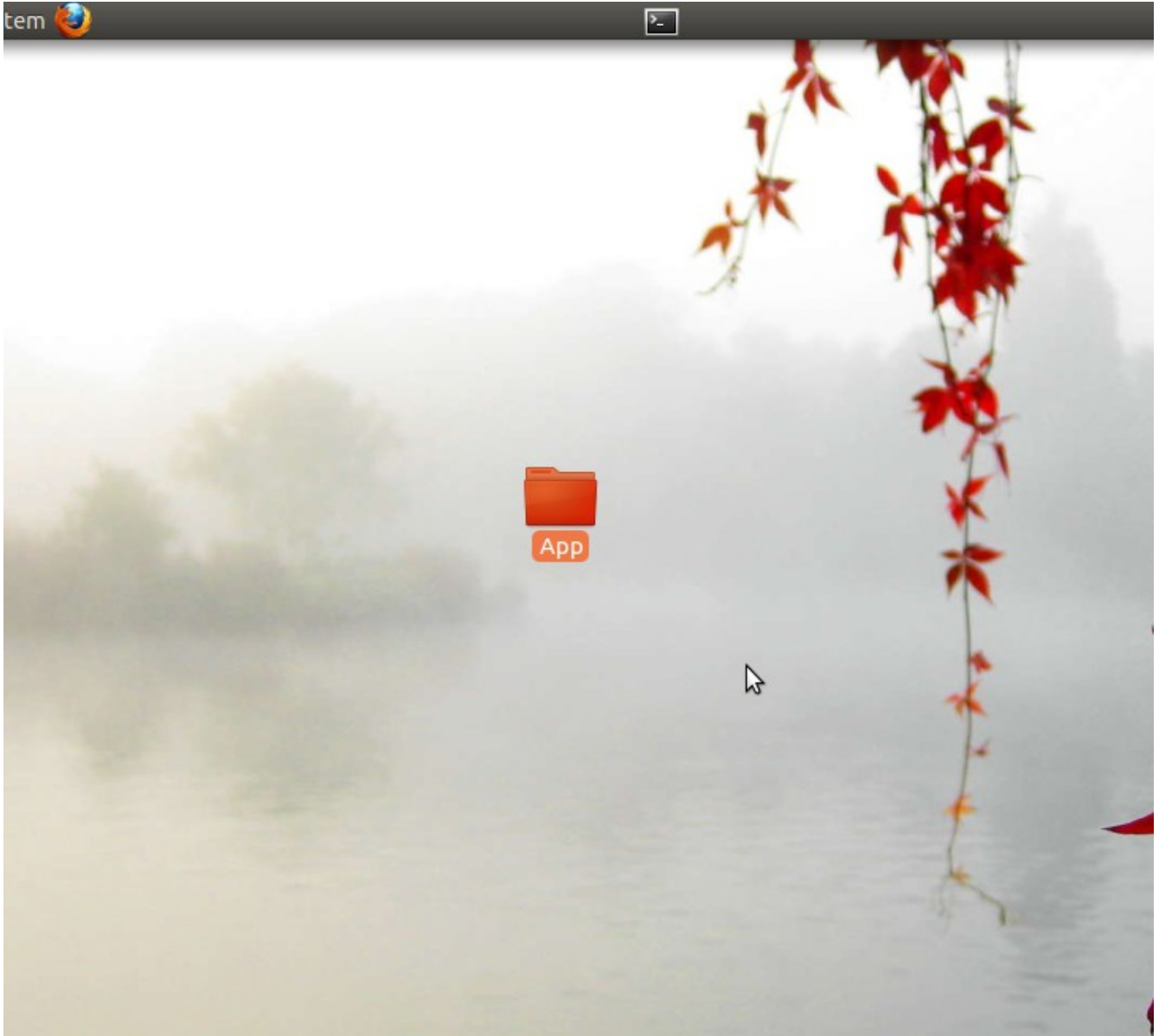
NetBeans IDE Download Bundles							
Supported technologies *	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
NetBeans Platform SDK	•	•	•				•
Java SE	•	•	•				•
JavaFX		•					•
Java Web and EE			•				•
Java ME			•				•
Java Card™ 3 Connected			—				—
Ruby				•			•
C/C++					•		•
Groovy			•				•
PHP						•	•
Bundled servers							
GlassFish Server Open Source Edition 3.0.1			•	•			•
Apache Tomcat 6.0.26			•				•
	<a href="#">Download</a> Free, 54 MB	<a href="#">Download</a> Free, 116 MB	<a href="#">Download</a> Free, 173 MB	<a href="#">Download</a> Free, 88 MB	<a href="#">Download</a> Free, 35 MB	<a href="#">Download</a> Free, 31 MB	<a href="#">Download</a> Free, 266 MB

ستقومون بإختيار خيار التنزيل للجافا فقط، فهي كافية ولا حاجة للمزيد من اللغات التي لا نحتاجها

هذه واجهة النت-بينز:



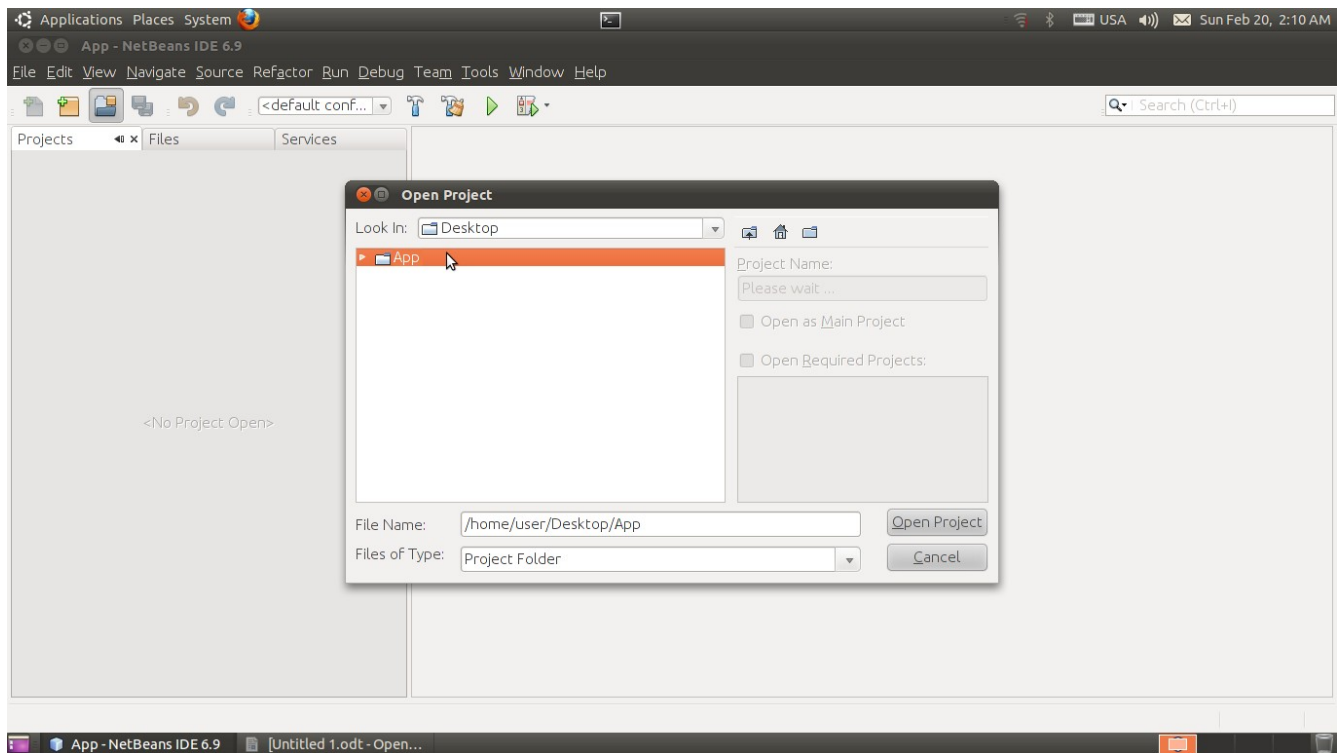
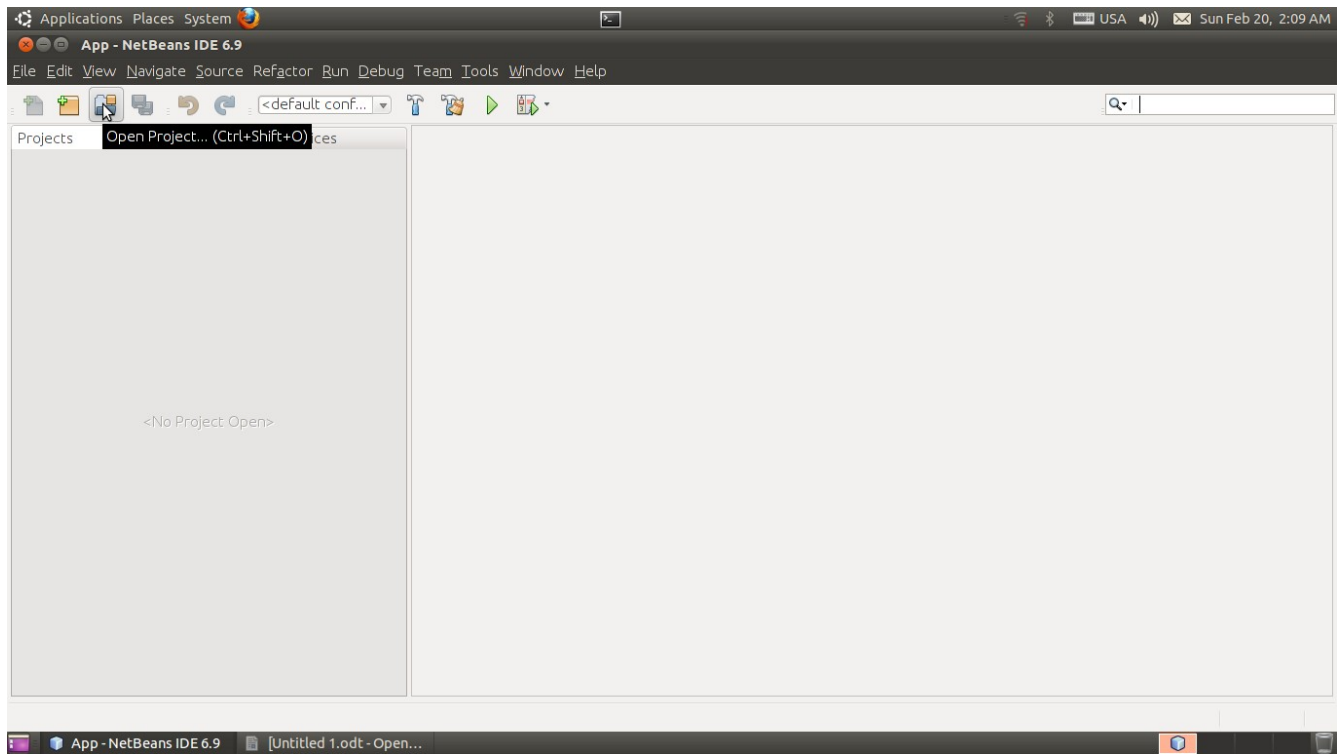
:البرنامج يأتي كملف مضغوط، تقومون بفك ضغط البرنامج، ومن ثم تستقبلون المجلد التالي



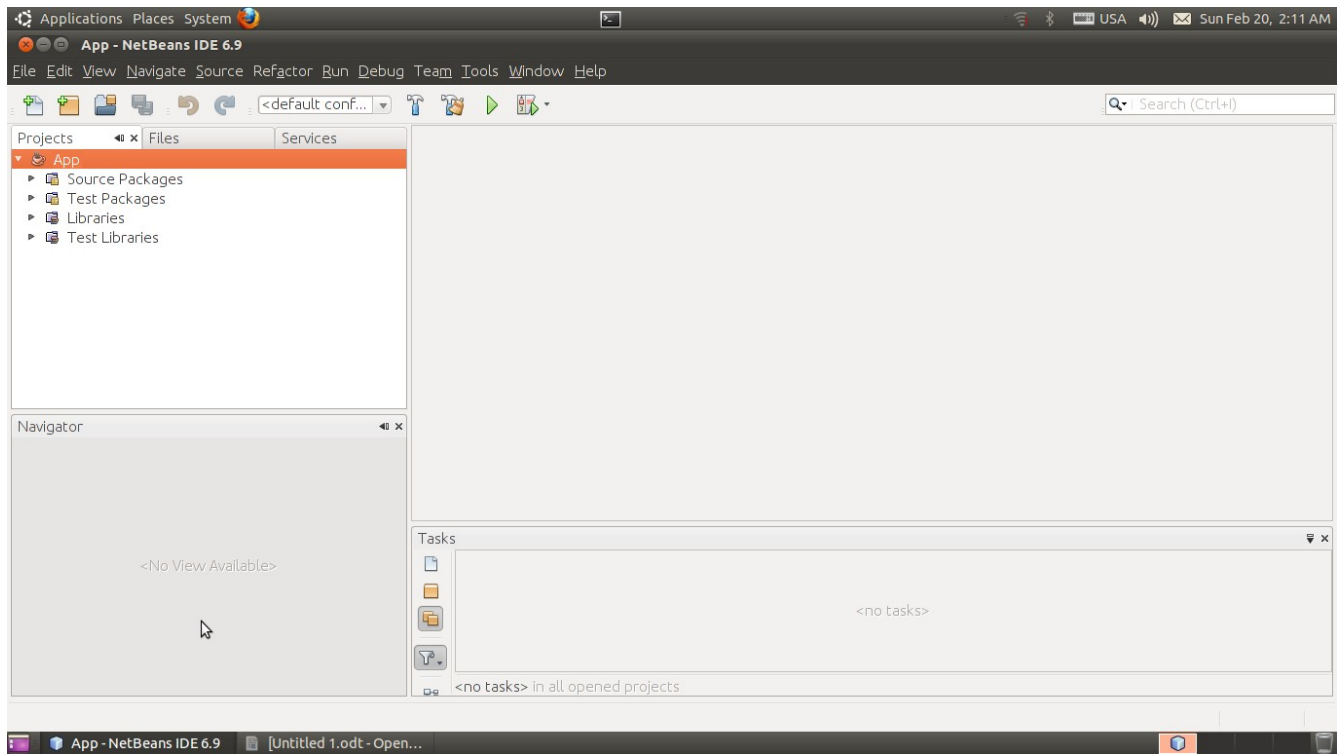
!هذا هو مجلد البرنامج

طبعًا البرنامج ليس جاهزًا للتشغيل، وهو معمول بالشكل الذي لو قامت المخابرات المرتدة بتنزيله فإنه لن ينفعهم البتة! فوضعنا لبرنامج تشفير يكون جاهزًا بشكل سهل، وتشغيله فقط، يُهدد إخواننا المجهدين المستخدمين لمثل هذا البرنامج! ولذلك فإننا سوف نطلعك على طريقة التعديل ليكون المشفر جاهزًا للعمل بإذن الله.

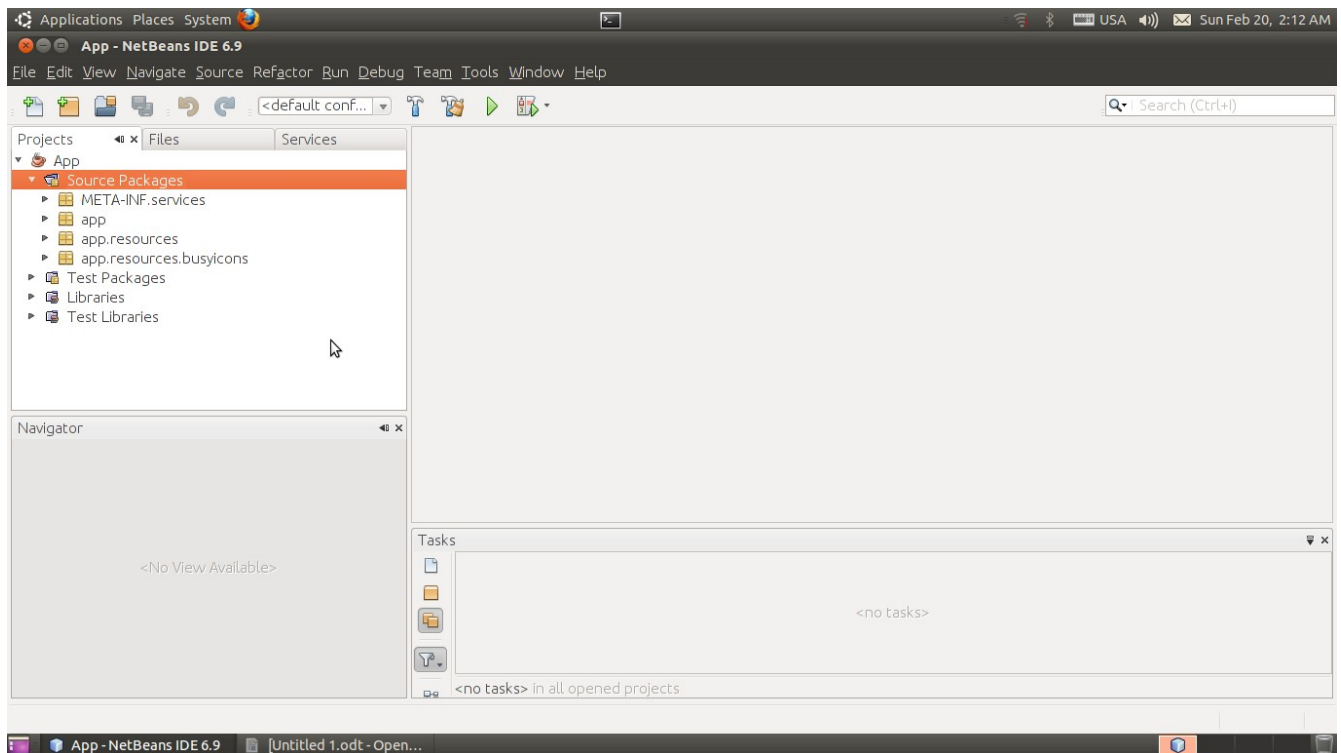
بعد قيامنا بتركيب النت-بينز للتعديل على البرنامج وبناءه، سوف نقوم برفع الملف على النت-بينز بإتباع الخطوات التالية:



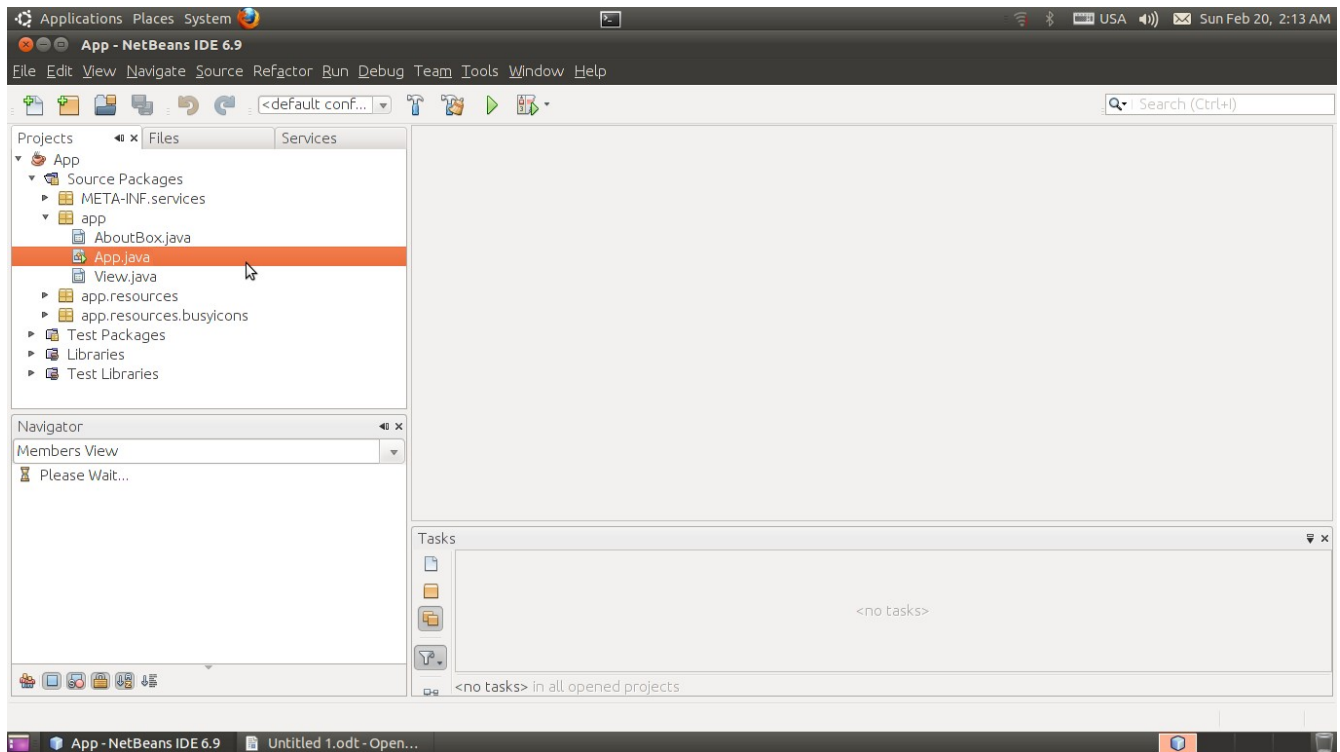
بعد الضغط على (فتح المشروع - البرويكت) تأتي الواجهة التالية



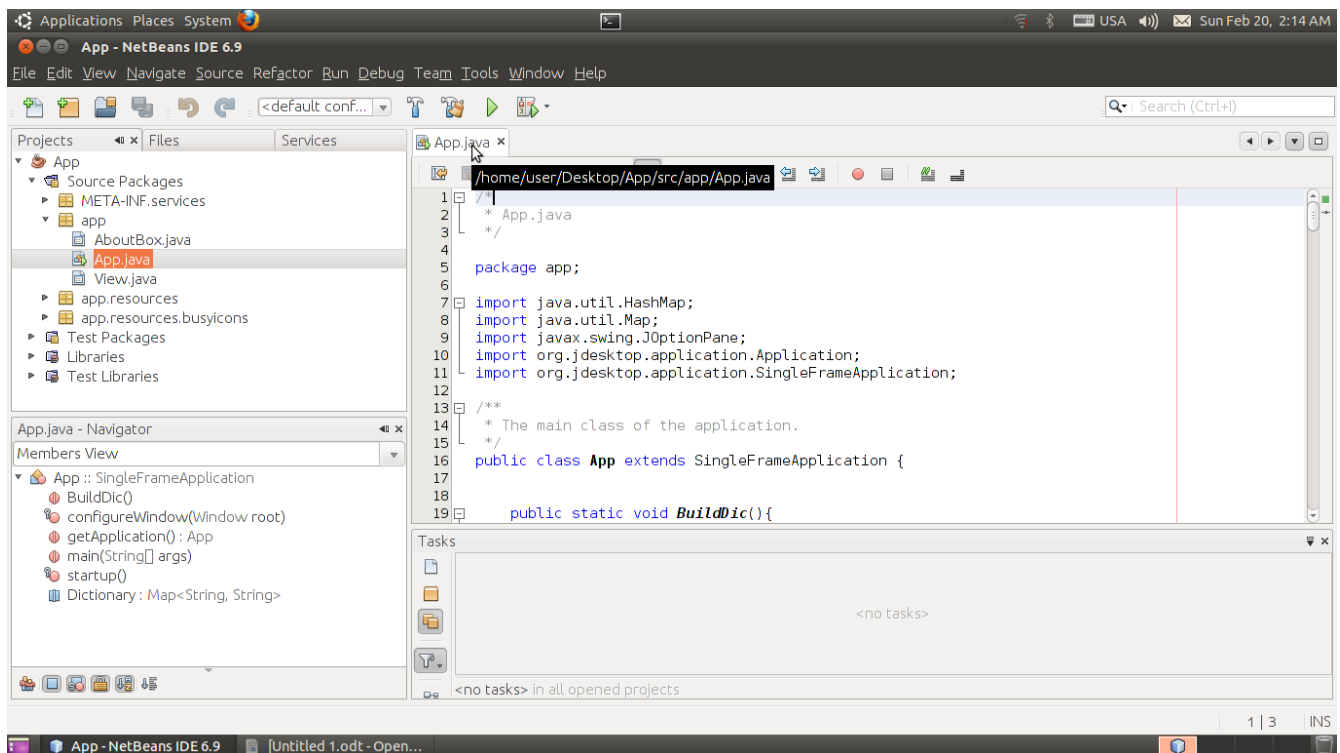
نقوم بالضغط مرتين على السورس كما هو ظاهر في الصورة التالية:



ومن ثم نضغط مرتين على آي-بي-بي كما في الصورة التالية:



ومن ثم نضغط على أي-بي-بي المعلمة وتأتي الواجهة التالية:

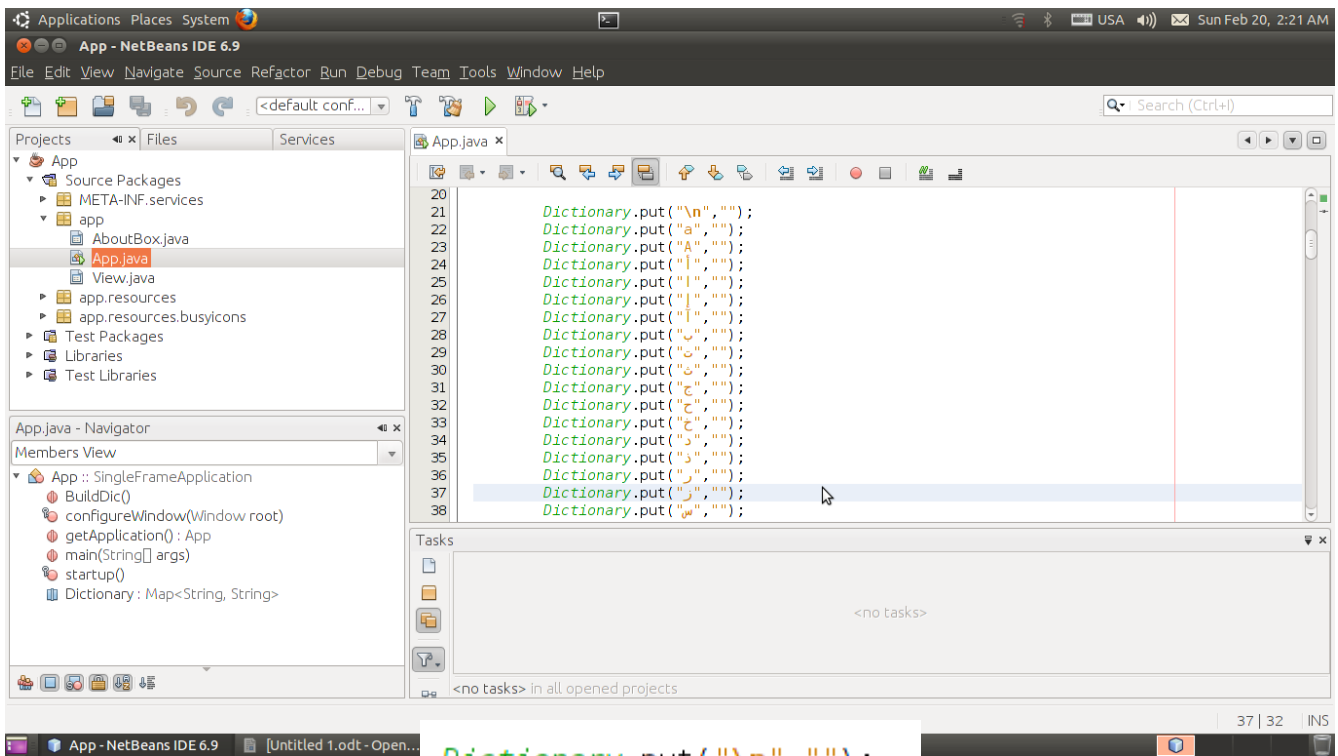


!!كما تشاهدون عرض برنامج النت-بينز السورس كود، أو الكود المصدري الأساسي للبرنامج قد يتسائل المستخدم، لماذا يظهر الكود المكتوب هكذا بكل سهولة، أي يسأل لماذا تم برمجة المشروع كأنه مشروع مفتوح المصدر؟  
الجواب: الأمر ليس كذلك! فهذه الخاصية من كون المشروع مفتوح المصدر، محدودة وسوف نقوم

- بإغلاق وحذف مصدريته للأبد! إنما أردنا من فتح الكود المصدري التالي
- 1- يمكن دراسة الكود وفحص إن كان فيه أكواد تجسس أو خبيثة
  - 2- يمكن تطوير برنامج التشفير للدارسين للغة الجافا
  - 3- يمكن إضافة إضافات جديدة على البرنامج واختيار خيارات أخرى مختلفة
  - 4- إعطاء المستخدم الأمن والحرية في اختيار الشيفرة التي يريد
  - 5- تحديد كلمات المرور للأمان والخاصية الأخرى

فالذي سوف يحصل، هو قيامنا باختيار الشيفرة بشكل يدوي! فوجود البرامج ذات الشيفرات المعروفة عالمياً يجعلها سهلة الفك، وذلك لأن الكود المصدري والأساسي الخاص بها متواجد في الأيدي، ويمكن دراسته وفهم كيفية عمله! ولكن بتشفيرنا اليدوي فإن احتمال فك التشفير هو ضئيل للغاية، وصعب جداً، إلا في عدة حالات سوف نذكرها في نقاط الضعف للبرنامج، وهي بالنهاية نقاط ضعف عند المستخدم لا البرنامج!

إذا نزلت إلى الأسطر الظاهرة في الصورة التالية



فإنه يظهر لكم وبشكل جلي، كيف أن كل رمز من اللغة العربية يُقابلة فراغ كما تشاهدون بشكل أوضح في الصورة التالية

```
Dictionary.put("\\n", "");
Dictionary.put("a", "");
Dictionary.put("A", "");
Dictionary.put("أ", "");
Dictionary.put("إ", "");
Dictionary.put("ا", "");
Dictionary.put("آ", "");
Dictionary.put("ب", "");
Dictionary.put("ت", "");
Dictionary.put("ث", "");
Dictionary.put("ج", "");
Dictionary.put("ح", "");
Dictionary.put("خ", "");
Dictionary.put("د", "");
Dictionary.put("ذ", "");
Dictionary.put("ر", "");
Dictionary.put("ز", "");
Dictionary.put("س", "");
```

ما سيقوم به المستخدم بدايةً، هو القيام بإعطاء كل رمز من الرموز، أو تشفير مقابل كل حرف، من الجدير بالذكر أنه يجب وضع 3 رموز سواءًا كانت حروفًا أو أرقامًا أو علامات تنقيط فقط! في حالة إعطاء البرنامج أكثر من 3 أو أقل فإنه لن يشتغل وسوف يخرج بأخطاء، ولن يخدمنا

:سأضع صورة لبعض الحروف كمثال لكيفية وضع التشفير

```
public static void BuildDic(){  
    Dictionary.put("\n", "");  
    Dictionary.put("a", "123");  
    Dictionary.put("A", "12e");  
    Dictionary.put("أ", "12E");  
    Dictionary.put("ا", "ghj");  
    Dictionary.put("إ", "bnm");  
    Dictionary.put("آ", "qaz");  
    Dictionary.put("ب", "wsx");  
    Dictionary.put("ت", "efv");  
    Dictionary.put("ث", ". , ?");  
    Dictionary.put("ج", " ; ' <");  
    Dictionary.put("ح", "");  
    Dictionary.put("خ", "");  
    Dictionary.put("د", "");  
    Dictionary.put("ذ", "");  
}
```

كما تشاهدون قمت بوضع ثلاث رموز معينة مقابل كل حرف، داخل أقواس التنصيص، وكما تشاهدون فإن الحروف من الحرف أ إلى الحرف ج مشفرة، والحرف ح و د و ذ ليسوا مشفرات، ولذلك يجب..ملئ كافة الحروف نهاية بالرموز والفواصل، إنتهاءً بالأرقام

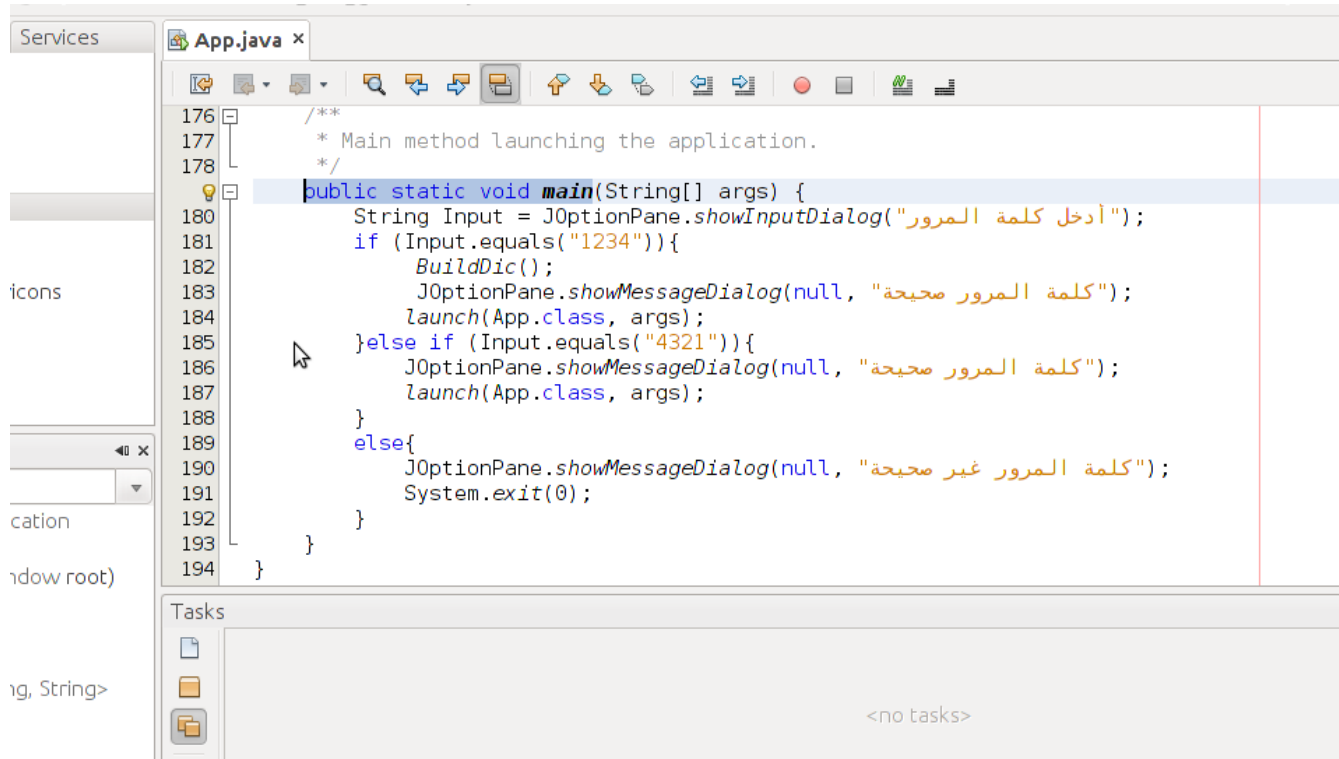
أنا وضعت العربية والإنجليزية للتشفير، لأنني قد أضع روابط معينة، أو كلمات إنجليزية

..ملاحظة هامة: يُنصح الدمج في التشفير بين الأحرف والأرقام والرموز  
وملاحظة هامة: هناك رموز معينة ممكن أن تخرب الكود، ولذلك أنصح بالرموز أعلاه المتواجدة مقابل



الحروف (ث و ج) كما تشاهدون في الصورة السابقة وملاحظة هامة: يجب أن يكون كل تشفير مكون من ثلاث خانات لكل حرف فريد، أي لا يجوز أن نقوم بعمل تشفير لحرفين بنفس القيمة التشفيرية!! ويمكن إدخال أحرف عربية، عبرية، صينية، في التشفير (: أيضًا، وذلك لأن برنامج نت-بينز يدعم اللغات ولله الحمد، كما يدعم العربية أيضًا

بعد ملئ كافة القائمة والرموز والأرقام، مقابل كل الحروف المراد تشفيرها، ننزل إلى تحت قليلًا في نفس الصفحة، ونجد التالي:

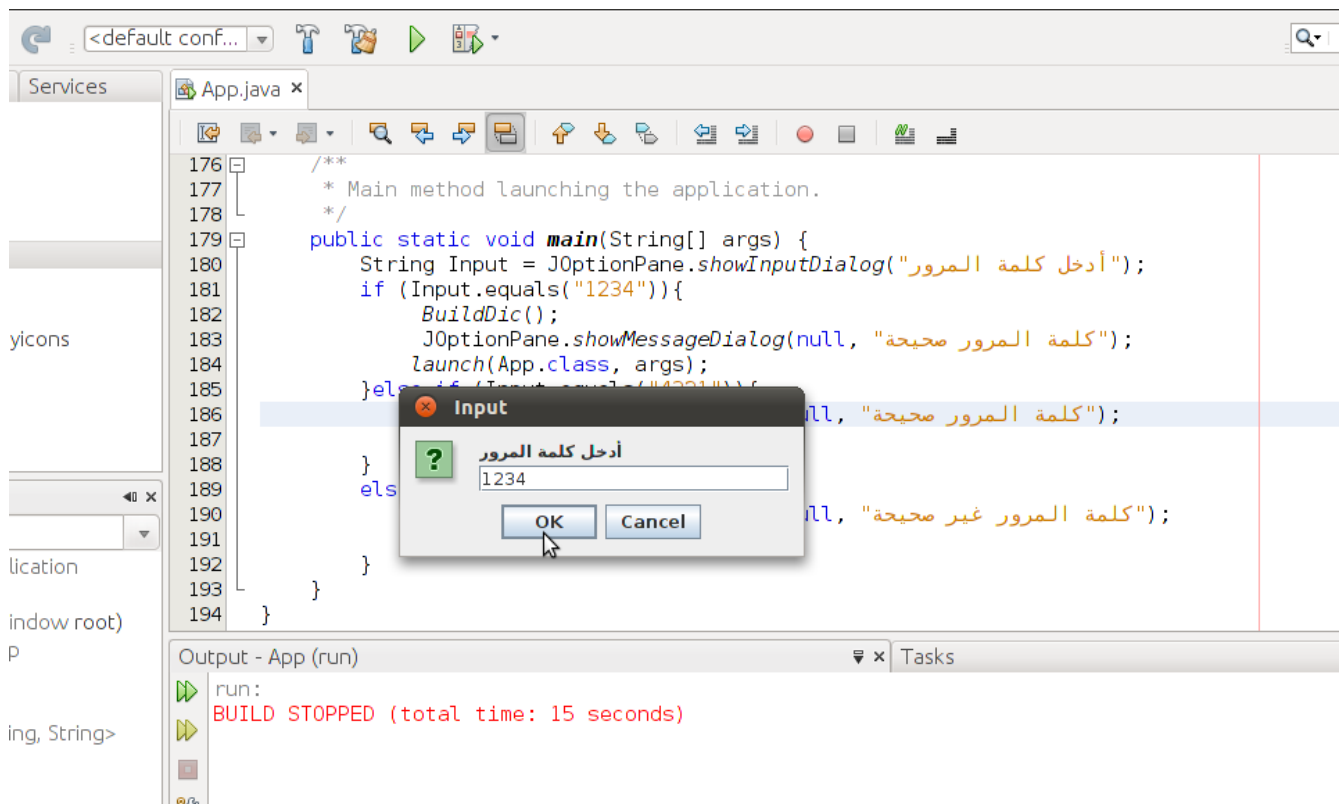


```
176 /**
177  * Main method launching the application.
178  */
179 public static void main(String[] args) {
180     String Input = JOptionPane.showInputDialog("أدخل كلمة المرور");
181     if (Input.equals("1234")){
182         BuildDic();
183         JOptionPane.showMessageDialog(null, "كلمة المرور صحيحة");
184         launch(App.class, args);
185     }else if (Input.equals("4321")){
186         JOptionPane.showMessageDialog(null, "كلمة المرور صحيحة");
187         launch(App.class, args);
188     }
189     else{
190         JOptionPane.showMessageDialog(null, "كلمة المرور غير صحيحة");
191         System.exit(0);
192     }
193 }
194 }
```

..إلى دالة تُسمى الممين، أي الدالة الأساسية، هنا سوف نقوم بعمل باسورد خاص ببرنامج التشفير إذا لاحظتم إن كان عندك معرفة صغيرة في الأكواد، أنا وضعت الرقم 1234 كباسورد للبرنامج، يقوم البرنامج بوضع قاموس التشفير وتشغيله عن طريق الدالة

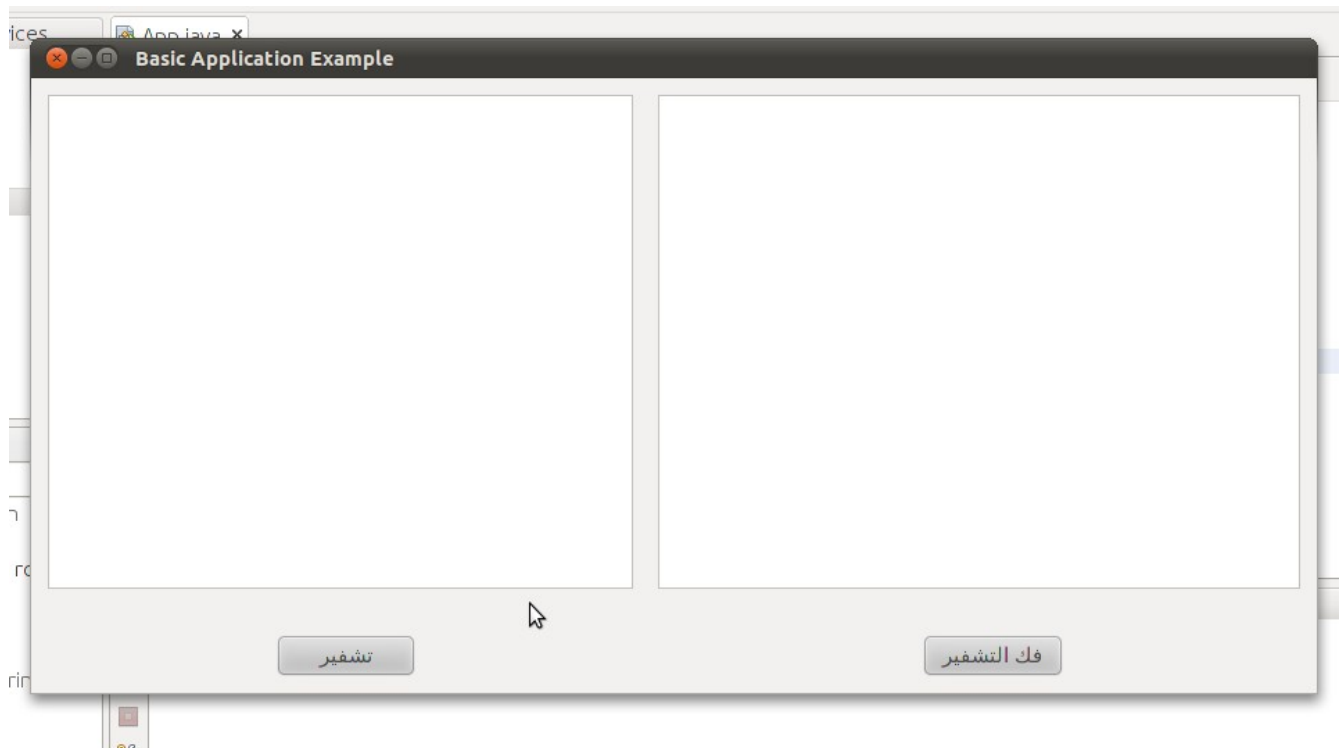
BuildDic

ومن ثم يظهر رسالة "كلمة المرور صحيحة" كما هو موضوع في الكود المصدري ومن ثم يشغل نافذة البرنامج والتي هي كالتالي:

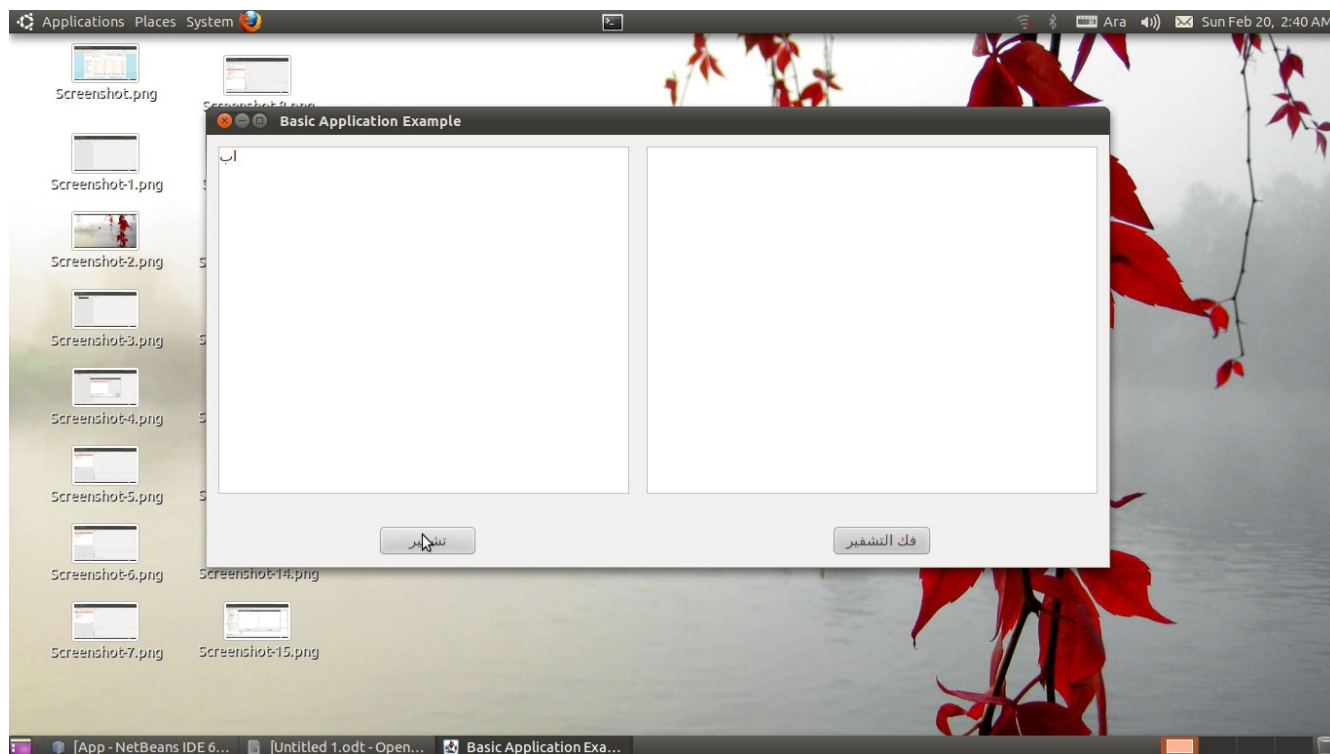


لا حاجة لذكر كيفية تشغيل الكود في هذه اللحظة، ولكنني أريد أن أريكُم مثلاً، تشغيل الكود يكون عن طريق الضغط على السهم الأخضر الواضح في الصورة السابقة، ولكن لا حاجة لمثل هذه

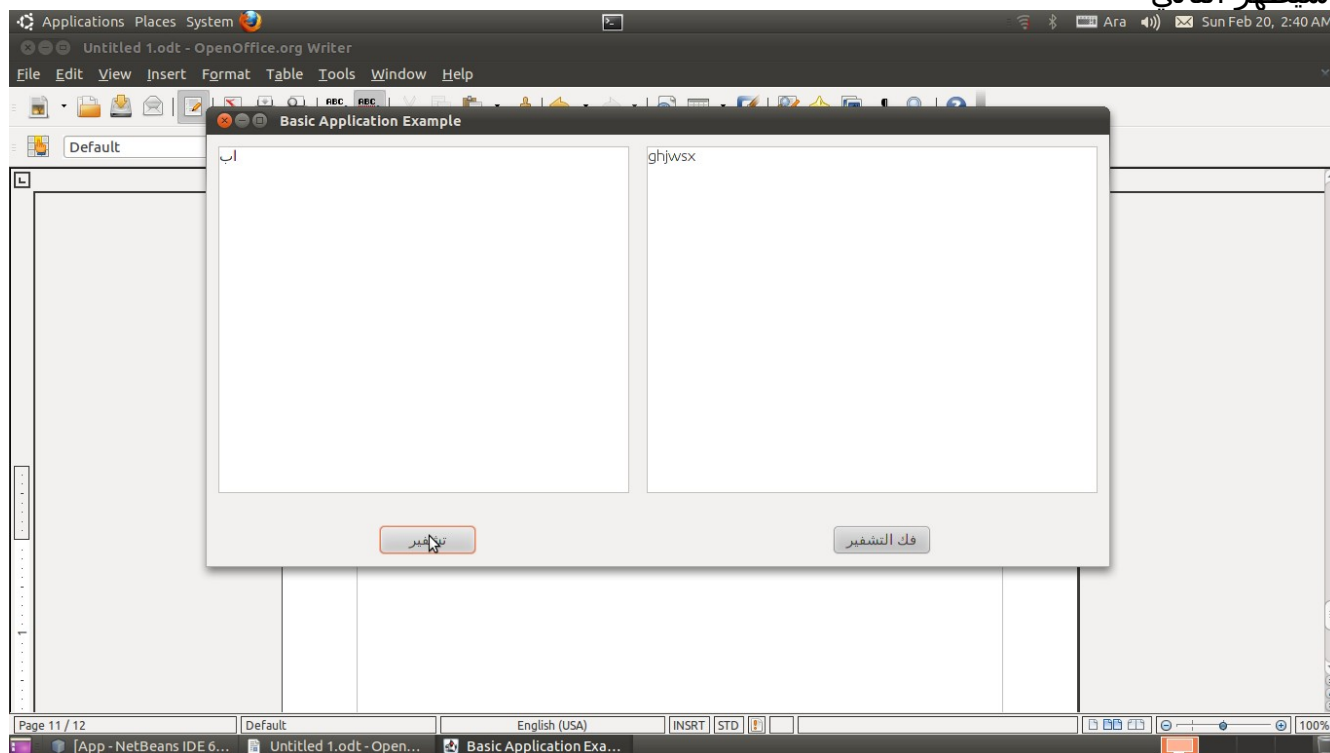
الآن بعد الدخول



كما تشاهدون هناك واجهة للتشفير وواجهة للفك، سوف نشفر مثالا



نضع كلمة اب ومن ثم نضغط على تشفير  
سيظهر التالي:



..وقد ظهر لنا التشفير في النافذة الأخرى.. والآن سوف نكمل

```

176  /**
177   * Main method launching the application.
178   */
179  public static void main(String[] args) {
180      String Input = JOptionPane.showInputDialog("أدخل كلمة المرور");
181      if (Input.equals("1234")){
182          BuildDic();
183          JOptionPane.showMessageDialog(null, "كلمة المرور صحيحة");
184          launch(App.class, args);
185      }else if (Input.equals("4321")){
186          JOptionPane.showMessageDialog(null, "كلمة المرور صحيحة");
187          launch(App.class, args);
188      }
189      else{
190          JOptionPane.showMessageDialog(null, "كلمة المرور غير صحيحة");
191          System.exit(0);
192      }
193  }
194  }

```

Output - App (run)

```

run:
Exception in thread "main" java.lang.NullPointerException
    at app.App.main(App.java:181)
Java Result: 1
BUILD SUCCESSFUL (total time: 2 seconds)

```

كما تشاهدون في الكود، وضعت كلمة المرور 4321 وأمرته بطباعة الأمر كلمة المرور صحيحة !! ولكن لماذا؟  
 الفرق أن الأمر الذي يشحن قاموس التشفير ليس موجودًا تحت أسطر هذا الباسوورد، مما يعني أن البرنامج لن يشتغل!! إبدأ ما الفائدة؟  
 الفائدة من هذه، عندما يتم إعتقال إخوة التوحيد، فإنه يستطيع فتح البرنامج للمخابرات، وإظهار أن النافذة كلمة المرور صحيحة تخرج، ولا يعمل البرنامج، مما سيظنون أن البرنامج تالف وليس صالحًا للإستعمال، وبعد الإنتهاء من جلسة المخابرات الملاحين، تدخل الباسوورد الأصل ويعمل البرنامج بشكل ممتاز :)، يمكن للدارس للغاية الجافا، القيام بإدخال باسوورد يقوم بحذف الحاسوب مثلًا ^\_\_^ هذه تعتمد على خبرة المطور للبرنامج، ولعلنا نقوم بإعطاء شروح عن كيفية إضافة أكواد تخدمك في وقت الصعائب!

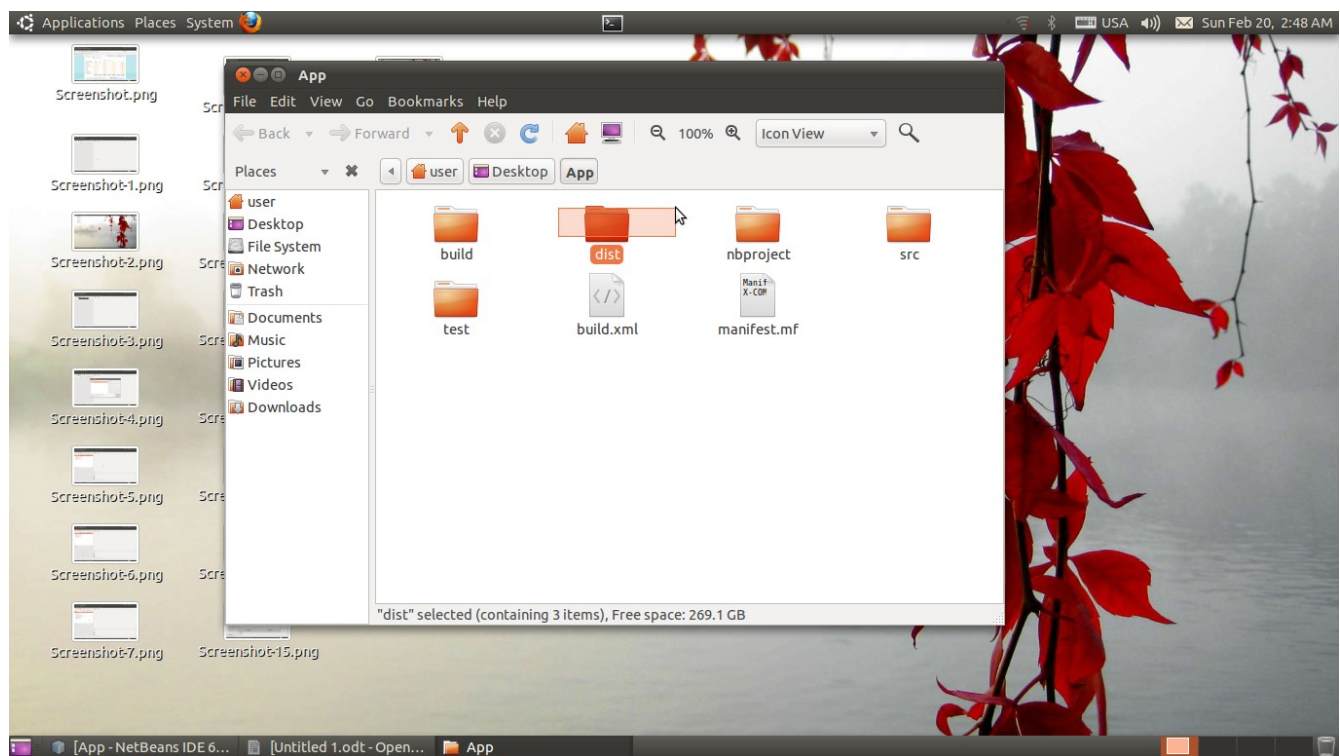
قم بتغير الباسوورد 1234 إلى الرقم أو الإسم الذي تُريد، تأكد منه جيدًا، قم بتغير الباسوورد الثاني 4321.. إلى الباسوورد الذي تُريد لخداع المرتدين بسهولة :) كما ظهر لكم سابقًا

والآن بعد الإنتهاء من وضع أكواد التشفير مقابل كل حرف من الحروف اللغات، ووضع الباسوورد المطلوب، نقوم بعمل التالي

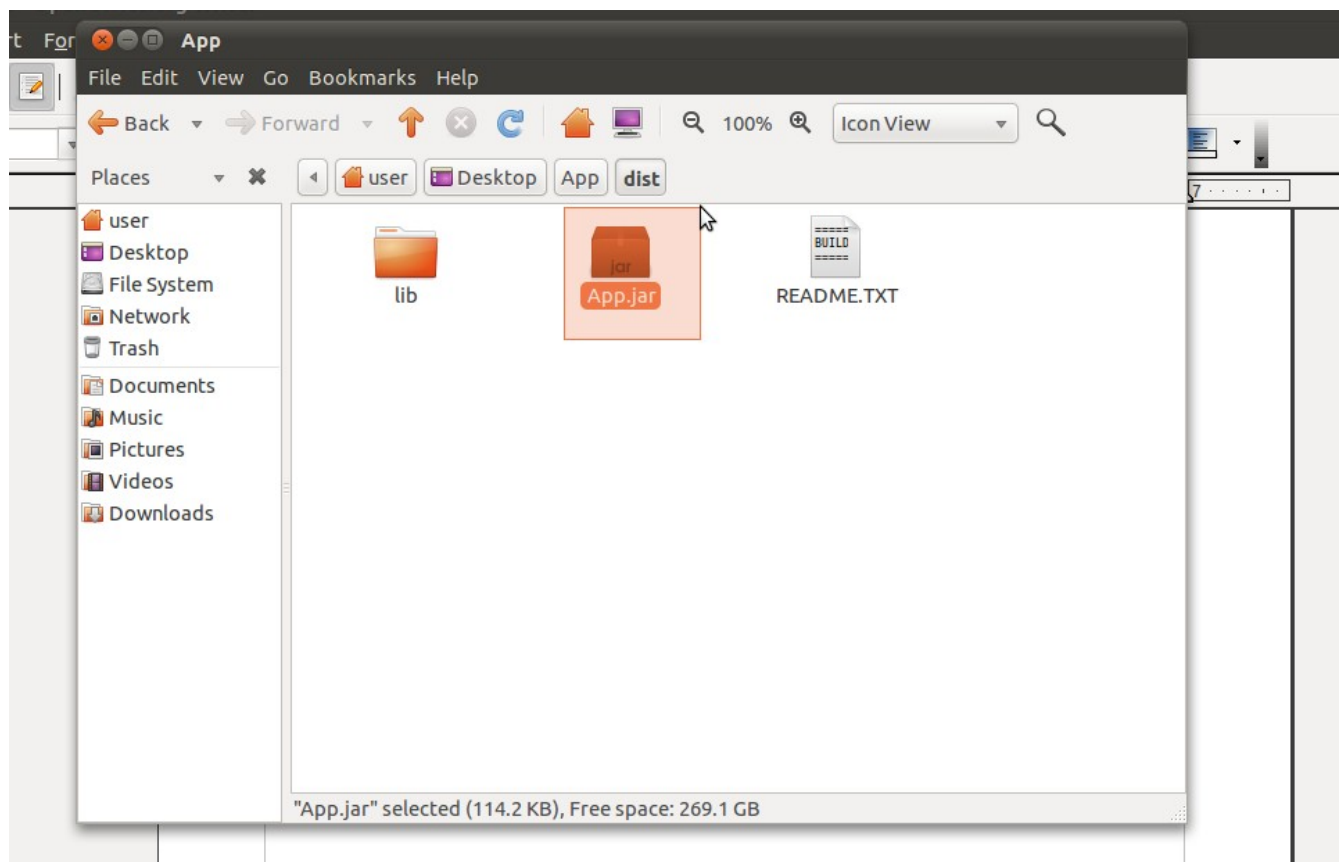
Shift + F11

سيقوم هذا الأمر ببناء ملفات التشغيل للبرنامج وسوف يقوم ببناء مجلد جديد داخل البرنامج بإسم dist

كما تشاهدون بالصورة



والآن ندخل على المجلد ديست



هذا الملف المُعلم بالفأرة والمسمى  
App.jar  
هو ملف التشغيل للبرنامج، ويختلف تشغيله حسب النظام

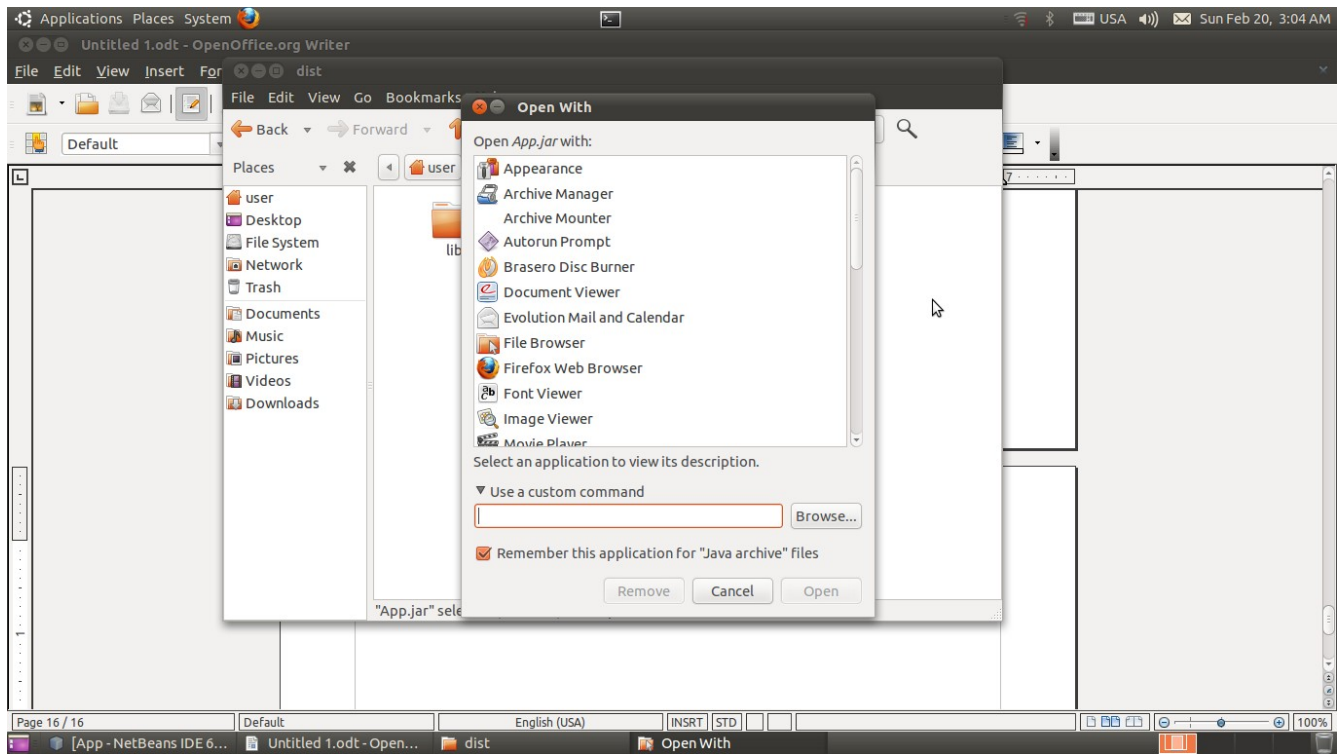
-لنظام ويندوز -سهل التشغيل  
نقوم بالضغط مرتين عليه

لنظام لينكس كافة التوزيعات نقوم بإعطاء الملف صلاحيات التشغيل، بالدخول إلى الترمينال، والذهاب  
إلى مكان الملف، بعد الوجود في مسار ملف الجار نقوم بالأمر التالي

```
r@root: ~/Desktop/App/dist
ew Search Terminal Help
~/Desktop/App/dist$ chmod 777 App.jar
```

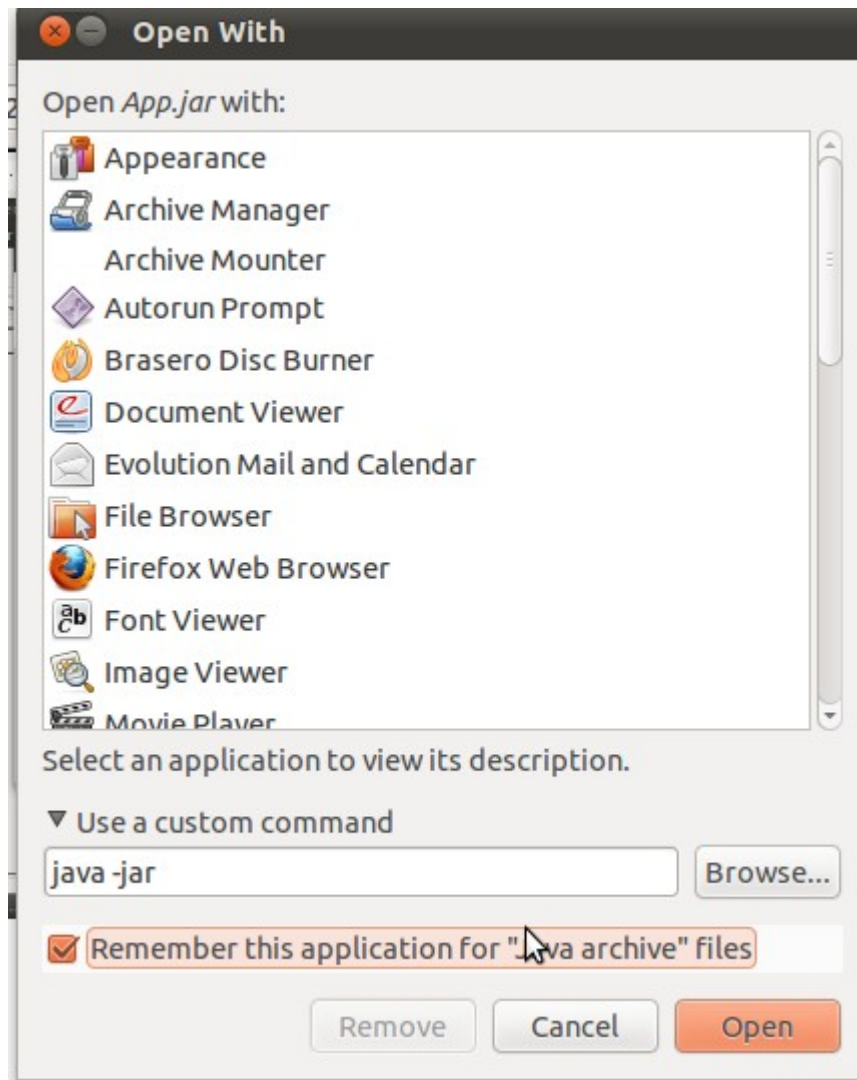
ونذهب للتشغيل كالتالي

نقوم بالضغط على الزر الأيمن من الفأرة على الملف وإختيار الخيار (فتح بواسطة): ومن ثم ستظهر  
الشاشة التالية



نقوم بالضغط على  
Use a custom command  
ومن ثم نكتب الأمر التالي  
java -jar  
كما هو ظاهر في الصورة التالية





ومن ثم عمل فتح، وبالتعليم على الخيار الذي يطأه الماوس، سوف يتم فتح البرنامج بمجرد الضغط مرتين على الملف، كما هو حال مستخدمى الويندوز

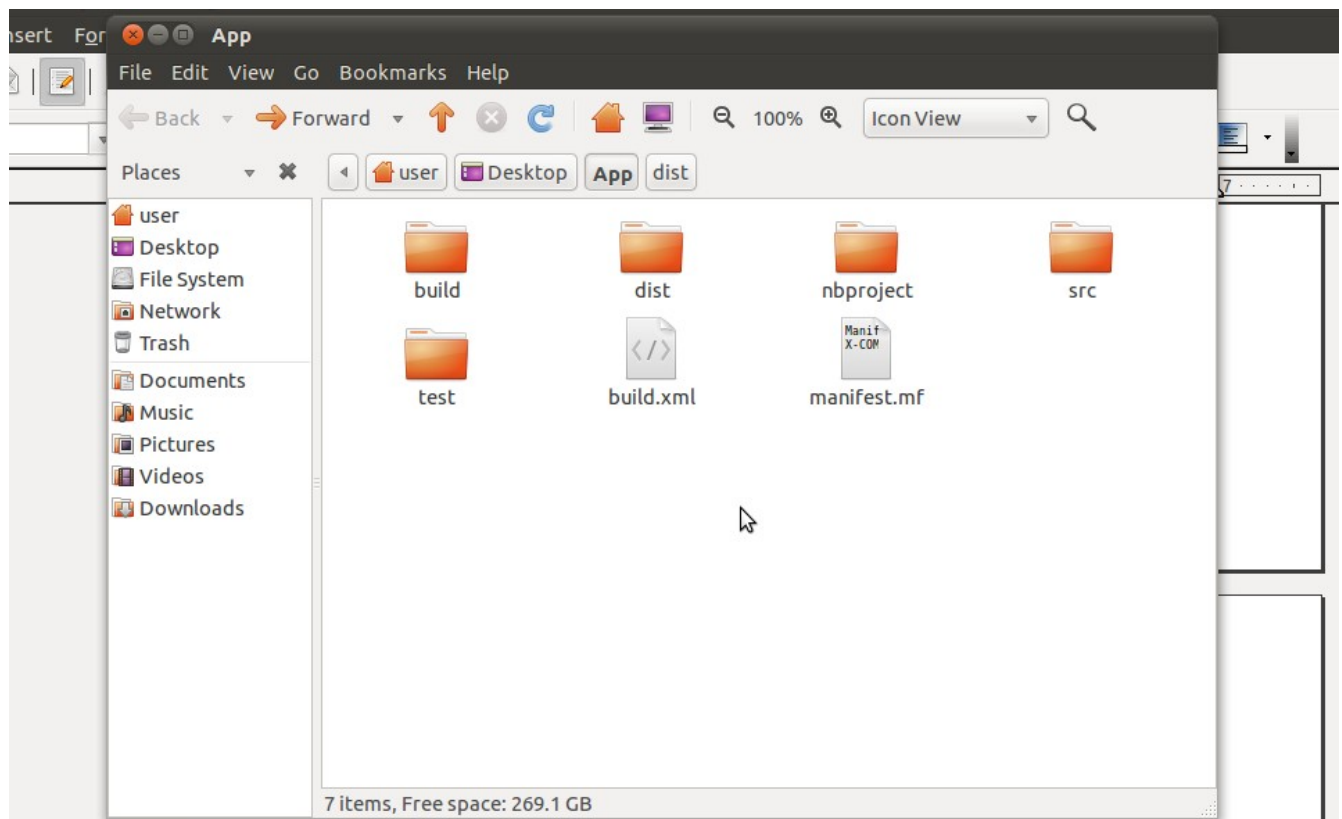
والآن بعد إشتغال البرنامج بشكل ممتاز، نقوم بإغلاق البرنامج، وعمل التالي  
طبعاً ذكرنا أن الكود المصدري مفتوح، وتستطيع حكومات الكفر أن تدخل إليه والنظر إلى التشفير،  
وتغير الباسوورد حتى!! ولذلك فسوف نقوم بالتالي  
سوف نذهب إلى مجلد المشروع، ونقوم بحذف المجلد

src

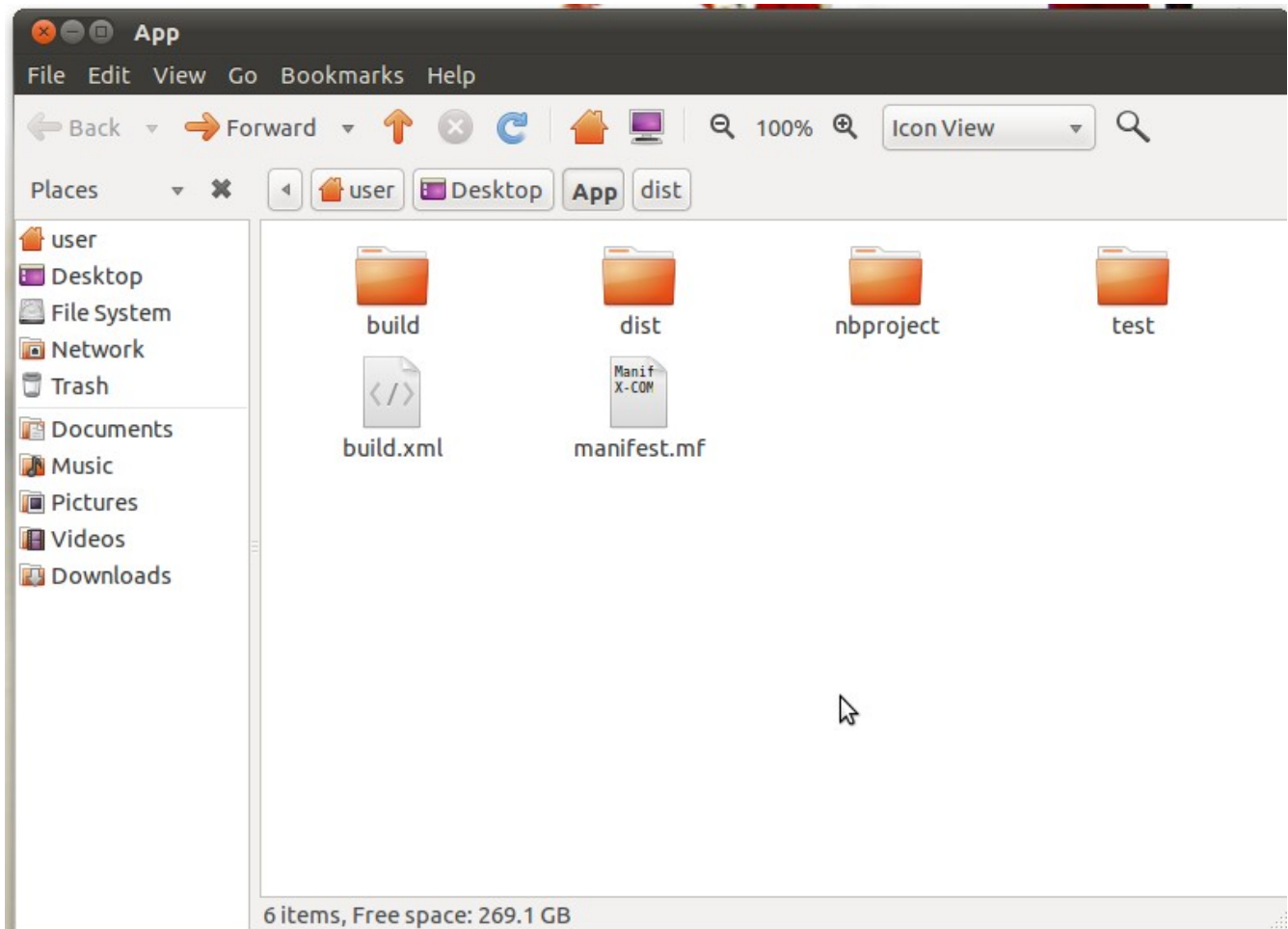
كما في الصورتين التاليتين

قبل





بعد



والآن الكود مغلق، والآن نريد التواصل مع الشخص ب، نقوم بإعطائه كامل المشروع، كما هو الآن بدون المجلد الذي حذف، ونعطي الشخص ب المجلد المعنون بآي-بي-بي بعد حذف مجلد السورس-كود طبقاً  
App

ونقوم بتعليمه طريقة تشغيل ملف "الجار" والذي سبق وذكرنا كيفية تشغيله

نقاط ضعف البرنامج

وضع رمز الضحك مثلاً (: هههههههه وذلك لأن التشفير سوف يبدو واضحاً أن الحرف نفسه كتب -1  
كثيراً، والأغلب أنه ضحك، فلا تكثرُوا الضحك فإنه يُميت القلب، وإبتسموا بأمر رسول الله صلى الله عليه وسلم

- 2- أن يقوم أحد الحاصلين على البرنامج بإعطائه لشخص آخر، فيستطيع الآخر تشفير الكلام، وهذه هامة
  - 3- فلا تعطي البرنامج إلا بمن تثق بهم، والله المستعان
  - 3- حاول قدر الإمكان إعطاء صاحبك باسورد فتح البرنامج والبرنامج بيدك لا عبر النت، لئلا يقع بأيدي
- الحكام المرتدين وجيوشهم الوثنية

إيجابيات البرنامج

- يدعم كامل لغات العالم بإذن الله سبحانه 1-  
مفتوح المصدر، يوفر الأمان بفحصه من التجسس، ويدعم التطوير للمشروع لمن يعرف لغة الجافا 2-  
سهل للغاية، وذو أمان عظيم لا يفك بعد التوكل على الله سبحانه 3-  
يُمكنك إضافة أي لغة، بوضع أحرفها، ووضع التشفير كما تم وضع اللغة العربية آنفًا، بنسخ السطر 4-  
الموجود للتشفير وهو  
Dictionary.put("Letter","Tashfer");  
راجع الصورة التالية

```
public static void BuildDic(){  
  
    Dictionary.put("\n","");  
    Dictionary.put("a","123");  
    Dictionary.put("A","12e");  
    Dictionary.put("أ","12E");  
    Dictionary.put("ا","ghj");  
    Dictionary.put("إ","bnm");  
    Dictionary.put("آ","qaz");  
    Dictionary.put("ب","wsx");  
    Dictionary.put("ت","efv");  
    Dictionary.put("ث",".,?");  
    Dictionary.put("ج",";'<");  
    Dictionary.put("ح","");  
    Dictionary.put("خ","");  
    Dictionary.put("د","");  
    Dictionary.put("ذ","");|
```

سوف نستقبل إستفساراتكم على منتديات شموخ الإسلام وعلى البريد التالي  
[muslimshamikh@yahoo.de](mailto:muslimshamikh@yahoo.de)

إنتهى الشرح  
والسلام عليكم ورحمة الله وبركاته

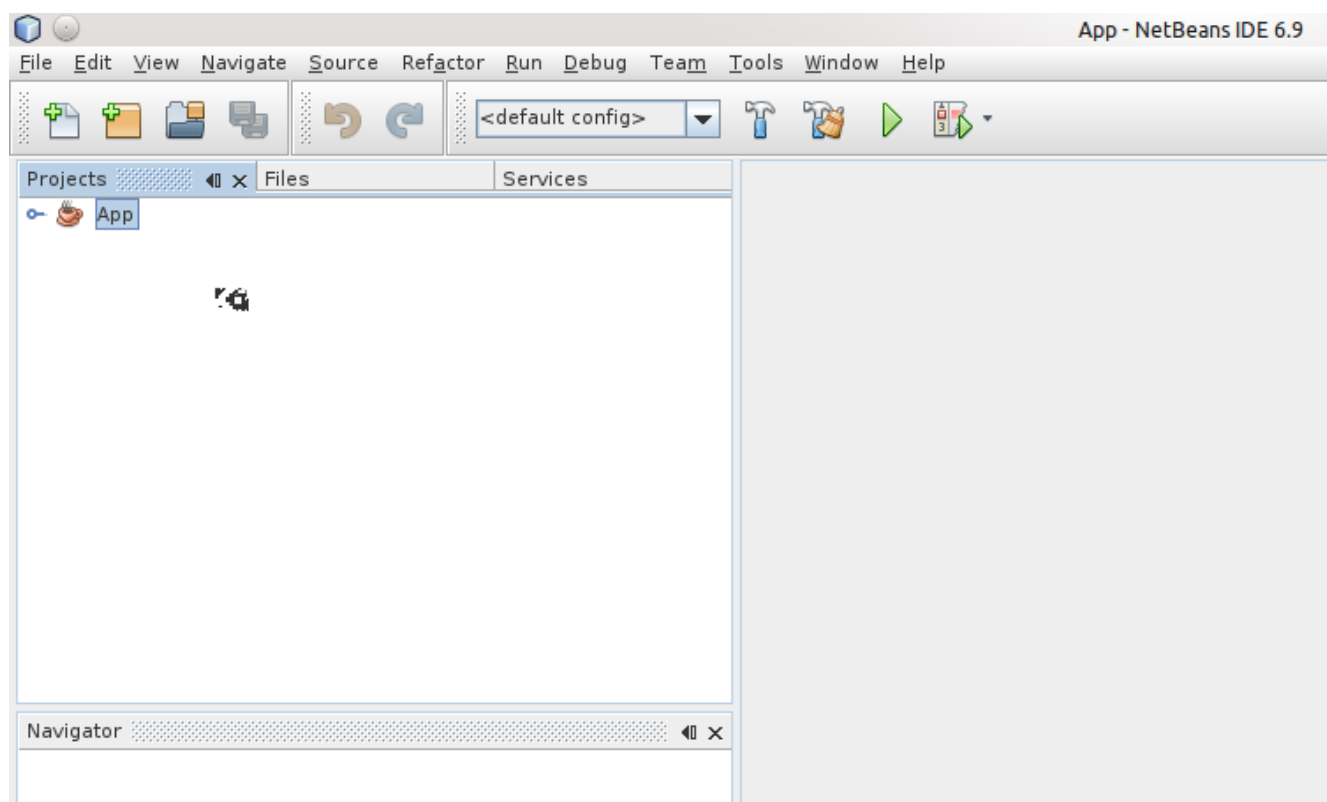
، السلام عليكم ورحمة الله وبركاته إخواننا الكرام  
هذا هو التحديث رقم "1"، في برنامج التشفير، والذي سوف يضمن التشفير التام والكامل في  
البرنامج، وكما تعلمون فإنه قد تم رفع البرنامج على كثير من المواقع، ولذلك فإنني لن أقوم بتحديث  
البرنامج كاملاً، ولكننا سوف نقوم ببعض الأمور والتي سوف تساعد في تطوير البرنامج إلى الأحسن  
بإذن الله سبحانه وتعالى، وسوف تكسر شوكة الصليبيين والعلماء المرتدين والحكام الخائنين لله  
ولرسوله -صلى الله عليه وسلم-، وهذا من بركة الله على أهل التوحيد والجهاد، ولله الحمد والمنة في  
ذلك! وسوف نعمل جاهدين على تطوير برامج المجاهدين حتى تكون أشد قوة ومثانة وأمنًا وسوف  
نُزودكم بكافة التحديثات اللازمة، والتطويرات للبرامج كافة، سواءً برنامج التشفير أو غيره بإذن الله،  
والعاقبة للمتقين.

**وأرى أن يتم إقتراح أفكار لبرامج لمحاولة دراسة إمكانية بناءها والعمل عليها، لنبني مجتمع برمجي آمن  
لإخواننا الموحدين في بقاع الأرض كافة**

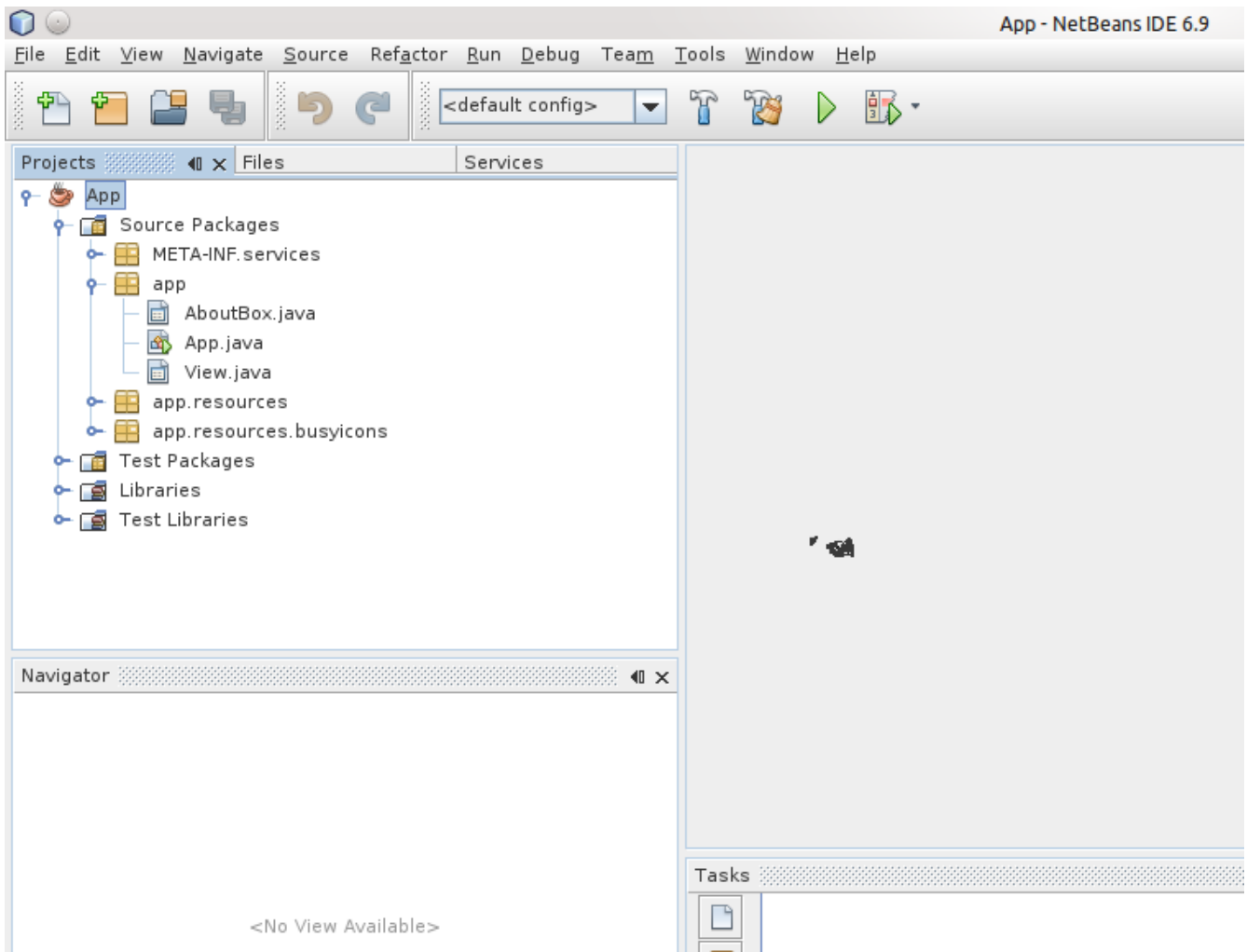
كما تعلمون فإننا حصرنا أسلوب التشفير في ثلاث خانات في شرحنا السابق، وهو الإصدار الأول من  
برنامج التشفير -بوضع ثلاث خانات مشفرة مقابل كل رمز من لوحة المفاتيح-، وبذلك فقد حصل  
العلماء المرتدون على معلومة وهي أن التشفير سيكون مكون من ثلاث رموز! ولذلك فمجرد إعطاءهم  
أدنى المعلومات، فإن ذلك سوف يساعدهم كثيرا في حربهم ضد الموحدين، ومحاولة إيجاد ثغرات في  
البرنامج والله المستعان، ولكن بعد التوكل على الله فالبرمجة بلغة الجافا قوية، والبرنامج متين صلب  
ولله الحمد والمنة، ونحن بإعتمادنا على مبدأ البرمجيات الحرة في برمجة كافة البرامج، فإن ذلك سوف  
يعطي المستخدم الأمان والحرية وإمكانية التطوير، وكنا قد ركزنا على هذا الأمر في الشرح الأول  
للإصدار الأول من البرنامج! فمربط الفرس والحل الأنجع أن تكون البرمجيات حرة، حيث يقوم  
المستخدم بتطبيق إعداداته وتطويره الذاتية بنفسه ومن دون تقييد، أضف إلى ذلك أن وضع برامج  
جاهزة وسهلة التشغيل، يهون على المرتدين تنزيلها وفك تشفير أي رسالة، ولذلك فإننا بحاجة إلى عون  
من المستخدم نفسه، ليتعلم كيفية الحماية القوية للغاية بعد التوكل على الله تعالى.

وبناءً على هذا المبدأ -مبدأ البرمجيات الحرة- فإننا سنشرح الآن طريقة لتصويب فك التشفير على  
المرتدين، زيادة إلى قدرة التشفير القوية، والتي من الصعب أن يفكها أعداء هذا الدين، وهي درس عن  
كيفية أن يقوم المستخدم بتحديد كمية خانات التشفير، وأن يقوم بتغيرها من ثلاث خانات إلى أربع أو  
خمسة أو ستة!! أو خانتين ولا أنصح بإستعمال خانة واحدة! وبهذا فلن يستطيع العدو تحديد كمية  
الخانات ليقترّب إلى الصورة أكثر، بل هذا يُوْهيه أكثر، فالجماعة "أ" تستعمل ثلاث خانات، والجماعة  
("ب" تستخدم خمس خانات، وهكذا... فأني له أن يعرف؟  
وليس الشرح والتطبيق صعبًا كما تظنون إذا تابعتم الشرح التالي بدقة إن شاء الله رب العالمين، وبسم  
الله نبدأ

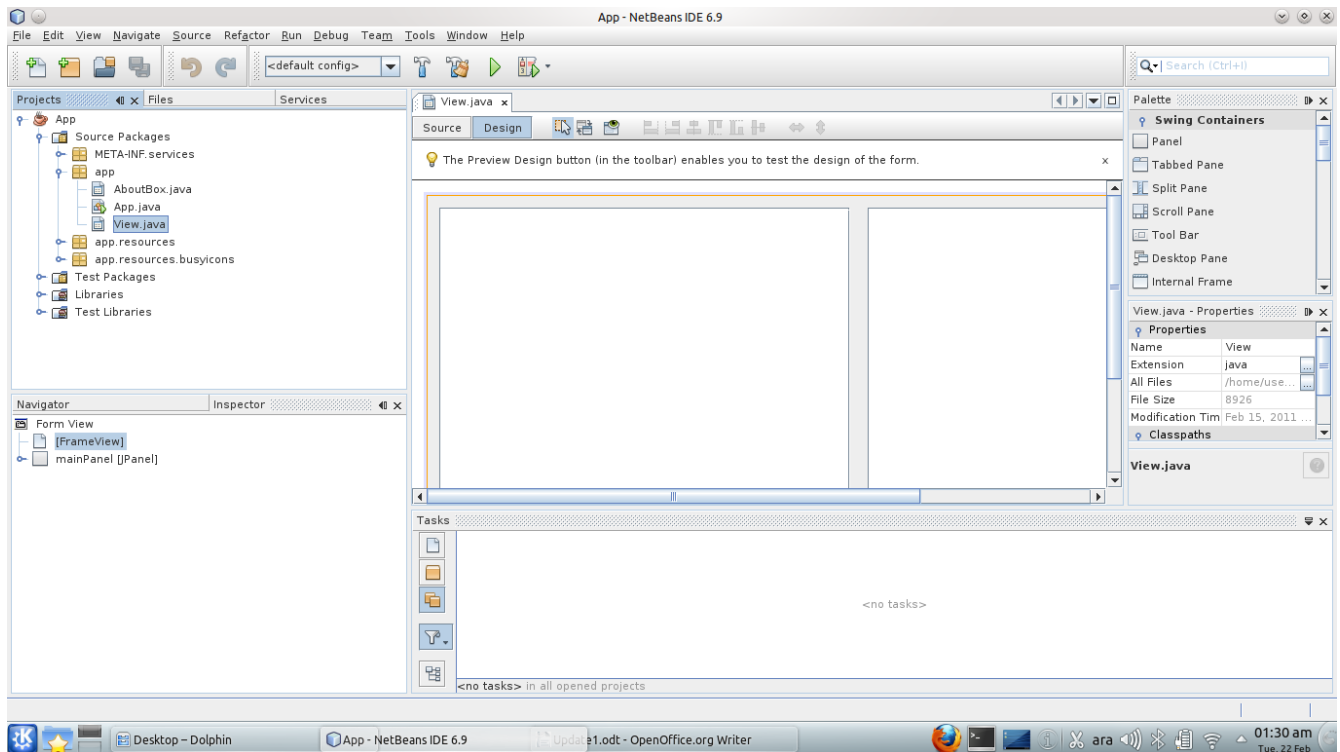
أولاً نقوم بالضغط على فتح المشروع لنقوم بالتعديلات والتطويرات المطلوبة، نقوم بالضغط على فتح  
مشروع، كما قمنا بذلك في شرح برنامج التشفير الإصدار الأول، حتى تظهر لنا النافذة التالية، مرفقة  
:- بمشروع الآي-بي-بي وهو برنامج التشفير -الذي قمتم بتنزيله سابقًا



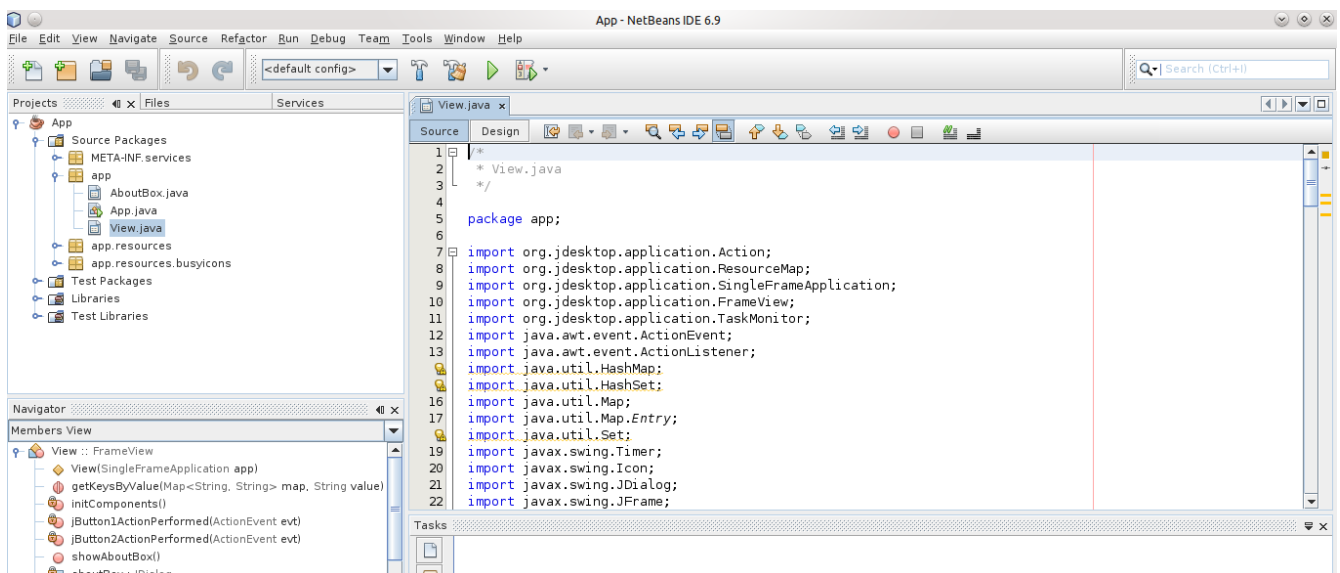
:ومن ثم نفتح المسارات الظاهرة في الصورة التالية وصولاً إلى الوضع التالي



الآن نريد أن نغير تحديد ثلاث خانات وجعلها أربع خاناتٍ مثلاً للتشفير!! فنقوم بالدخول إلى الملف  
View.java  
بالضغط مرتين على إسم الملف، حتى تظهر لنا الصورة التالية:



نقوم بتحديد الخيار الذي يقوم بعرض الكود المصدري للبرنامج وهو بالضغط على  
Source  
وهو الآن مضغوط على  
Design  
حتى يظهر الكود، وهو موجود في النافذة اليمينية كما في الصورة التالية



الآن نقوم بالنزول إلى السطور التالية كما في الصورة

```
152
153 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
154     //Decrypt Button
155     String In = jTextArea1.getText();
156     jTextArea2.setText("");
157     char[] vArray = In.toCharArray();
158     int flag = 0;
159     String temp = "";
160     for (Character v:vArray){
161         JOptionPane.showMessageDialog(null,temp);
162         if (flag == 3){
163
164             flag = 0;
165             String Update = getKeysByValue(App.Dictionary,temp);
166             jTextArea2.append(Update);
167             temp = "";
168
169         }
170         temp+=v.toString();
171         flag++;
172     }
```

هذا هو الكود المصدري الذي يقوم بالتشفير.. الآن نقوم بتغيير الرقم 3 في السطر التالي  
if (flag == 3)

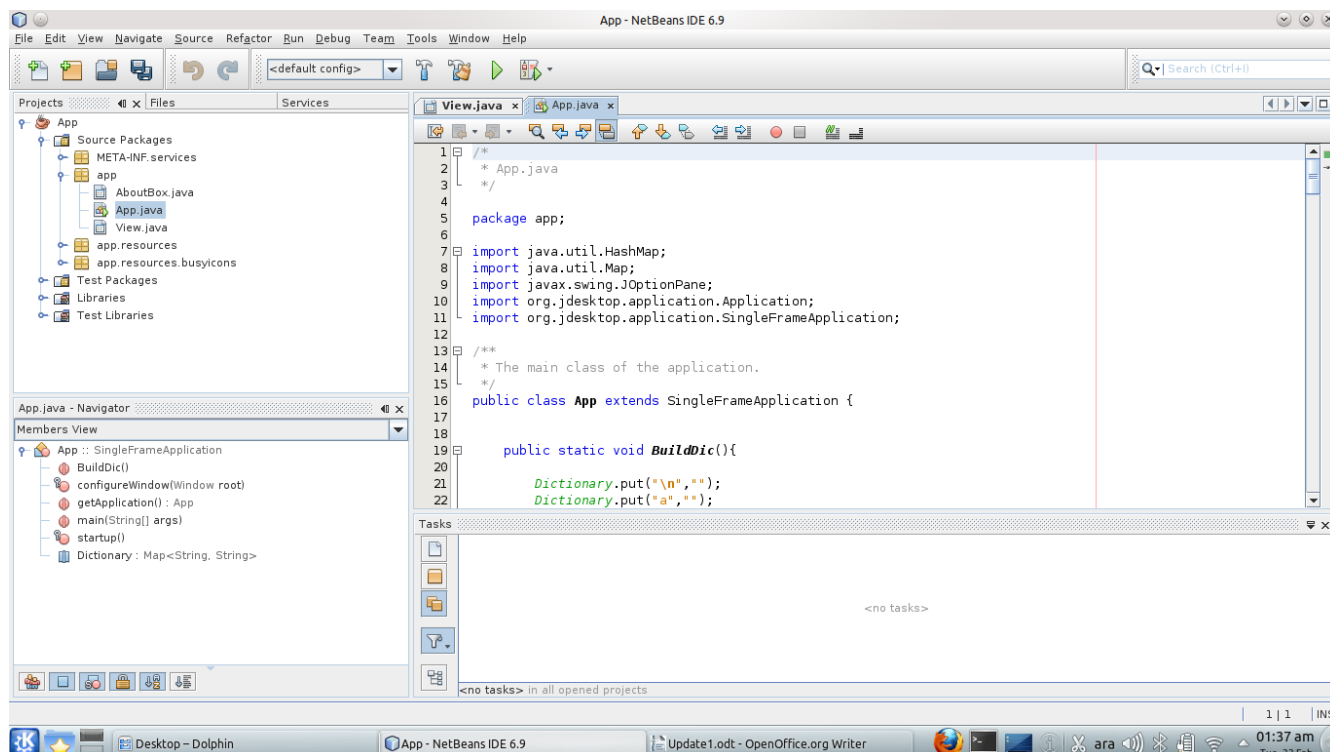
ونضع عدد خانات التشفير المراد وضعها.. مثلاً لو وضعنا أربع خانات سيكون الكود كالتالي

```
152
153 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
154     //Decrypt Button
155     String In = jTextArea1.getText();
156     jTextArea2.setText("");
157     char[] vArray = In.toCharArray();
158     int flag = 0;
159     String temp = "";
160     for (Character v:vArray){
161         JOptionPane.showMessageDialog(null,temp);
162         if (flag == 4){
163
164             flag = 0;
165             String Update = getKeysByValue(App.Dictionary,temp);
166             jTextArea2.append(Update);
167             temp = "";
168
169         }
170         temp+=v.toString();
171         flag++;
172     }
```

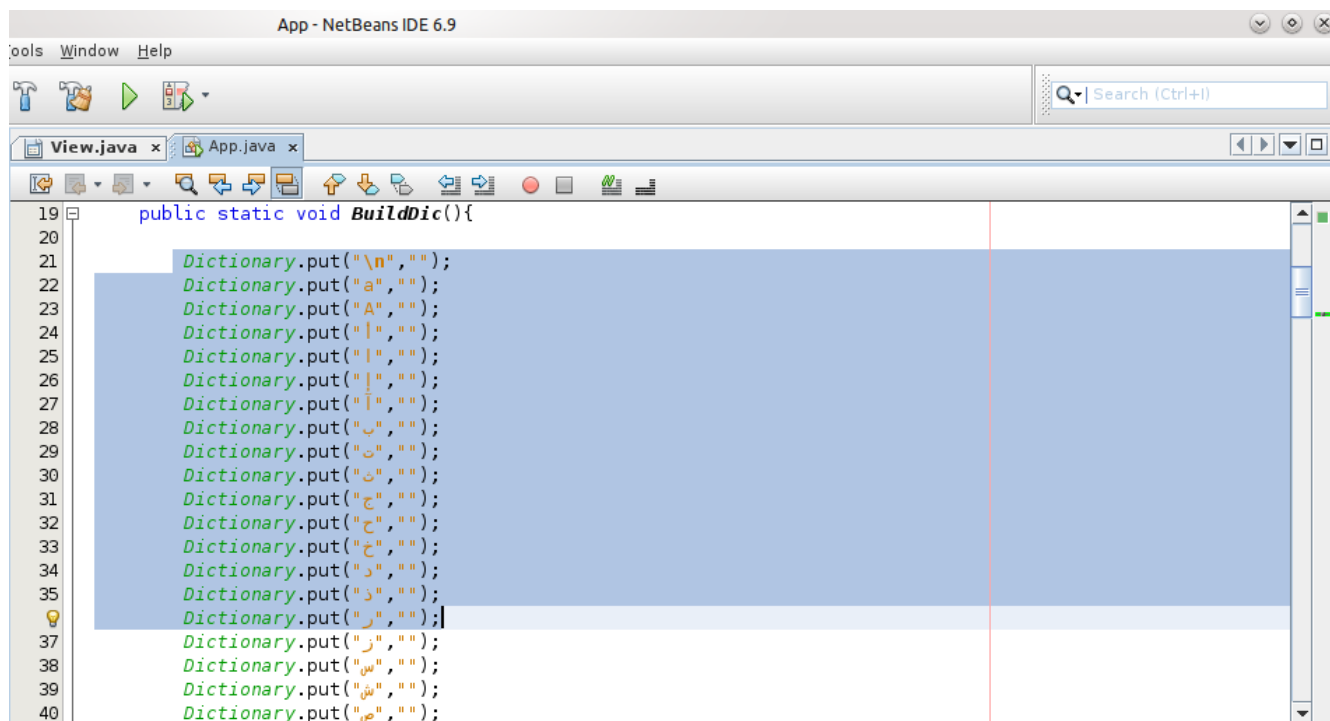
والآن سوف نتجه إلى الملف الذي فيه قاموس التشفير وهو في الملف  
App.java



نقوم بالدخول عليه بالضغط مرتين في النافذة الشمالية كما في الصورة التالية



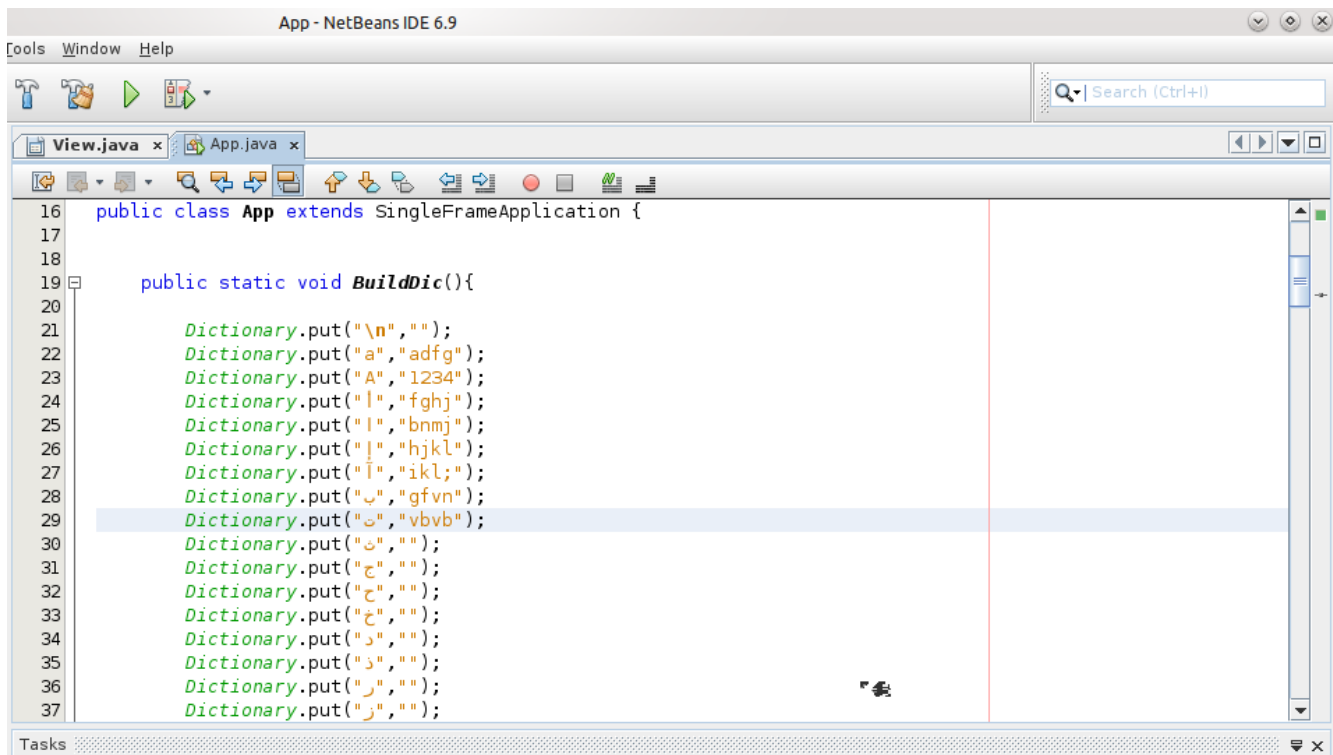
والآن نقوم بإعطاء كل رمز موجود في القاموس قيمة مكونة من أربع خانات لا ثلاث خانات، لتغير عدد خانات التشفير كما في الصورة التالية



لن أشرح كيفية تعبئة قاموس التشفير، فهي موجودة في الشرح الأولي لبرنامج التشفير الإصدار الأول!! وهكذا فقد تغير عدد خانات التشفير

والسؤال: هل هذا يعني أنه كلما زادت خانات التشفير كلما زاد الأمان؟  
والجواب: نستطيع القول بأنه كلما زادت خانات التشفير كلما زاد الأمان، ولكن لا قلق من وضع ثلاث خانات تشفير أو أكثر، فذلك آمن بما يكفي لصعوبة فك التشفير من قبل المرتدين أعداء هذا الدين، ولكن بإعطاء المستخدم طريقة تغير خانات التشفير فإن هذا سوف يترك الكفرة الملاحين، وسوف يصعب عليهم كثيرا عملية فك التشفير الشبه مستحيلة ولله الحمد والمنة في ذلك

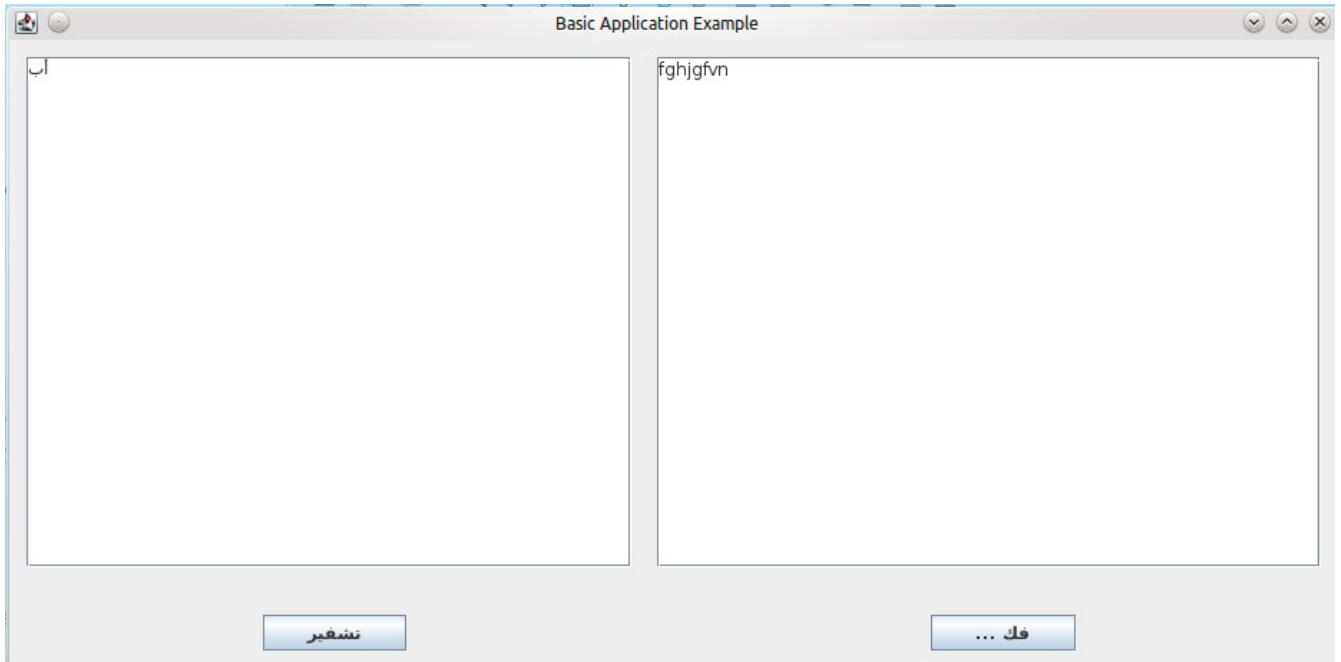
طبعاً بعد الإنتهاء من التعبئة بشكل كامل، نقوم بعمل بناء لملفات التشغيل كما شرحنا في الشرح الأولي للبرنامج، وضعت بعض التشافير في الصورة التالية بغية فحصها



```
16 public class App extends JFrameApplication {
17
18
19     public static void BuildDic(){
20
21         Dictionary.put("\n", "");
22         Dictionary.put("a", "adfg");
23         Dictionary.put("A", "1234");
24         Dictionary.put("l", "fghj");
25         Dictionary.put("l", "bnmj");
26         Dictionary.put("l", "h j k l");
27         Dictionary.put("I", "i k l");
28         Dictionary.put("ب", "gfvn");
29         Dictionary.put("ت", "v b v b");
30         Dictionary.put("ت", "");
31         Dictionary.put("ع", "");
32         Dictionary.put("ح", "");
33         Dictionary.put("ح", "");
34         Dictionary.put("د", "");
35         Dictionary.put("ز", "");
36         Dictionary.put("ر", "");
37         Dictionary.put("ز", "");
```

سوف نقوم بعمل بناء لملفات التشغيل بالضغط على  
Shift+F11

ندخل ملف الجار ونحاول تشغيله وتجريب تشفير الكلمة أب، والتي تحتوي على الحروف "أ" و "ب"  
..والتي قمنا بعمل تشفير لها للفحص



!ويظهر بأن خانات التشفير أصبحت أربعة، وتعمل كما يجب وبشكل إعتيادي وممتاز

الهدف من هذا التطوير في عملية التشفير: هو محاولة لإرباك المرتدين، فكل جماعة تختار عدد خانات التشفير الخاص بها، وتقوم بوضع التشفير الخاص بها في "قاموس التشفير" وبهذا فإن الإحتمالات سوف تكبر وسوف يكون البرنامج أكثر أمانًا والحمد لله رب العالمين

لمزيد من الإستفسار  
زوروا منتديات شموخ الإسلام - قسم منتدى أمن الإتصال وأنظمة تشغيل لينكس  
أو على البريد التالي  
muslimshamikh@yahoo.de